

Approximation Algorithms for Path TSP, ATSP, and TAP via Relaxations

by

Zhihan Gao

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2015

© Zhihan Gao 2015

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The results in Chapters 3 and 4 are based on my single-authored papers [33], [34]. The results in Chapter 5 is based on the paper [15] co-authored with Joseph Cheriyan, Konstantinos Georgiou, and Sahil Singla. The results in Chapters 6 and 7 are based on the papers [12], [13] co-authored with my supervisor.

Abstract

Linear programming (LP) relaxations provide a powerful technique to design approximation algorithms for combinatorial optimization problems. In the first part of the thesis, we study the metric s - t path Traveling Salesman Problem (TSP) via LP relaxations.

We first consider the s - t path graph-TSP, a critical special case of the metric s - t path TSP. We design a new simple LP-based algorithm for the s - t path graph-TSP that achieves the best known approximation factor of 1.5. Then, we turn our attention to the general metric s - t path TSP. [An, Kleinberg, and Shmoys, STOC 2012] improved on the long standing $\frac{5}{3}$ -approximation factor and presented an algorithm that achieves an approximation factor of $\frac{1+\sqrt{5}}{2} \approx 1.61803$. Later, [Sebő, IPCO 2013] further improved the approximation factor to $\frac{8}{5}$. We present a simple, self-contained analysis that unifies both results. Additionally, we compare two different LP relaxations of the s - t path TSP, namely the path version of the Held-Karp LP relaxation for TSP and a weaker LP relaxation, and we show that both LPs have the same (fractional) optimal value. Also, we show that the minimum cost of integral solutions of the two LPs are within a factor of $\frac{3}{2}$ of each other. Furthermore, we prove that a half-integral solution of the stronger LP relaxation of cost c can be rounded to an integral solution of cost at most $\frac{3}{2}c$. Finally, we give an instance that presents obstructions to two natural methods that aim for an approximation factor of $\frac{3}{2}$.

The Sherali-Adams (SA) system and the Lasserre (Las) system are two popular Lift-and-Project systems that tighten a given LP relaxation in a systematic way. In the second part of the thesis, we study the Asymmetric Traveling Salesman Problem (ATSP) and unweighted Tree Augmentation Problem, respectively, in the framework of the SA system and the Las system.

For ATSP, our focus is on negative results. For any fixed integer $t \geq 0$ and small ϵ , $0 < \epsilon \ll 1$, we prove that the integrality ratio for level t of the SA system starting with the standard LP relaxation of ATSP is at least $1 + \frac{1-\epsilon}{2t+3}$. For a further relaxation of ATSP called the balanced LP relaxation, we obtain an integrality ratio lower bound of $1 + \frac{1-\epsilon}{t+1}$ for level t of the SA system. Also, our results for the standard LP relaxation extend to the path version of ATSP.

For the unweighted Tree Augmentation Problem, our focus is on positive results. We study this problem via the Las system. We prove an upper bound of $(1.5 + \epsilon)$ on the integrality ratio of a semidefinite programming (SDP) relaxation, where $\epsilon > 0$ can be any small constant, by analyzing a combinatorial algorithm. This SDP relaxation is derived by applying the Las system to an initial LP relaxation. We generalize the combinatorial analysis of integral solutions from the previous literature to fractional solutions by identifying some properties of fractional solutions of the Las system via the decomposition result of [Karlin, Mathieu, and Nguyen, IPCO 2011].

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor Prof. Joseph Cheriyan for his invaluable guidance and continued support for my research during the Ph.D. program. When I need advice and help, he is always there. He taught me a lot about how to write a scientific paper, how to select a suitable research topic, and, most importantly, how to be a good researcher. Without these, I would never have been able to complete this thesis.

Besides my supervisor, I would also like to thank the other members of the thesis examining committee for reading my thesis and giving useful comments: Prof. Timothy Chan, Prof. William Cook, Prof. Anupam Gupta, and Prof. Levent Tunçel.

My special thanks go to my research collaborators for the stimulating discussions in our research meetings: Zachary Friggstad, Konstantinos Georgiou, and Sahil Singla.

I would like to thank my colleagues in the C & O department for providing suggestions and encouragement for my research: Prof. Chaitanya Swamy, Prof. Laura Sanità, Andre Linhares and Abbas Mehrabian.

Many thanks to my friends in Waterloo, who make my life colorful even in the hard times of my Ph.D. journey: Ahmad Abdi, Fidel Barrera-Cruz, Jianfa Cong, Gabriel Gauthier-Shalom, Feng He, Junbo Huang, Mehdi Karimi, Nishad Kothari, Hadi Minooei, Chunlin Wang, Siying Wei, Miaolan Xie, Kun Xiong, Xin Xiong, and Hanmeng Zhan.

Last but not least, I would like to thank my family. I would not have made it so far without their support. I thank my mother, Mei Li, and my father, Bohai Gao, for their unconditional love and care. I thank my wife, Ye Wang, for staying with me through all the ups and downs of life. I thank my grandmother Wenxiu Duan for holding my tiny hands and telling what the outside world looks like when I was a kid. She will live forever in my heart.

Dedication

To the memory of my dear grandmother, Wenxiu Duan.

Table of Contents

List of Figures	xi
1 Introduction	1
2 Preliminaries	7
2.1 LP relaxations	7
2.1.1 Path TSP	8
2.1.2 Asymmetric TSP	10
2.1.3 TAP	11
2.2 Lift-and-Project systems	11
3 Path Graph Traveling Salesman Problem	16
3.1 Preliminaries	17
3.2 LP-based $\frac{3}{2}$ -approximation algorithm	18
4 Path Traveling Salesman Problem	22
4.1 Preliminaries	23
4.1.1 Linear programs	24
4.1.2 T -joins	25
4.1.3 Polyhedra and convex decomposition	25
4.1.4 Christofides' algorithm for s - t path TSP	26

4.2	Unified correction vector	26
4.2.1	AKS' $\frac{1+\sqrt{5}}{2}$ -approximation via unified correction vector	30
4.2.2	Sebő's $\frac{8}{5}$ -approximation via unified correction vector	31
4.3	Linear programming relaxations of the s - t path TSP	33
4.4	Counterexample to two approaches	38
5	On Integrality Ratios for Asymmetric TSP in the Sherali-Adams System	43
5.1	Our results	44
5.2	Preliminaries	46
5.2.1	LP relaxations for Asymmetric TSP	46
5.2.2	The Sherali-Adams system	47
5.3	SA applied to the Balanced LP relaxation of ATSP	50
5.3.1	Certifying a feasible solution	50
5.3.2	CGK (Charikar-Goemans-Karloff) construction	56
5.4	SA applied to the standard LP (DFJ LP) relaxation of ATSP	60
5.4.1	Certifying a feasible solution	63
5.5	Path ATSP	69
6	Approximating (Unweighted) Tree Augmentation via Lasserre System, Part I: Stemless TAP	73
6.1	Our results and techniques	74
6.2	Preliminaries and notation	77
6.3	The initial LP	80
6.4	Lasserre tightening and its properties	81
6.5	Potential function for stemless TAP	84
6.6	Algorithm	86
6.6.1	Semiclosed trees	87
6.6.2	Credit assignment	88

6.6.3	Simple contractions and assertions on M	88
6.6.4	Good semiclosed trees	89
6.6.5	Algorithm in summary	91
6.6.6	Worked example	91
6.7	Analysis of the algorithm	92
6.7.1	Semiclosed trees are good except deficient 3-leaf trees	93
6.7.2	Addressing deficient 3-leaf trees	97
6.8	Tight example for the analysis	99
7	Approximating (Unweighted) Tree Augmentation via Lasserre System, Part II	102
7.1	Preliminaries and notation	104
7.2	Lasserre tightening and its properties	106
7.3	Potential function	107
7.4	Algorithm and credits I: Preprocessing steps	112
7.4.1	Semiclosed trees	113
7.4.2	Maximum matching	114
7.4.3	Bad 2-stem trees	114
7.4.4	Credit assignment for the algorithm and the preprocessing	119
7.4.5	Second preprocessing step	122
7.5	Algorithm and credits II: Overall algorithm	122
7.5.1	(Up-to-5) greedy contractions and assertions on M	122
7.5.2	Good semiclosed trees	123
7.5.3	Summary of the algorithm	124
7.5.4	Stem assertion of the algorithm	124
7.6	Analysis of the algorithm, and deficient trees	125
7.6.1	Properties from assertions	127
7.6.2	Most semiclosed trees are good	130
7.6.3	Addressing deficient trees	141

8	Conclusion	144
8.1	Path TSP	144
8.2	Asymmetric TSP	145
8.3	TAP	146
	References	147

List of Figures

2.1	A comparison of the Lift-and-Project systems. $A \longrightarrow B$: B is stronger than A . .	14
4.1	Tight example	37
4.2	Support graph of x^{H_b} with edge x^{H_b} values and edge costs	39
4.3	Cost of the good spanning tree	41
4.4	Tree J_b	41
5.1	A digraph G with a good decomposition given by the dicycle with thick edges, and the length 2 dicycles C_j formed by the anti-parallel pairs of thin edges; $G - E(C_j)$ is strongly connected for each dicycle C_j	50
5.2	G_0 and G_1 for $r = 3$	56
5.3	G_k and L_k for $k \geq 2$ and $r = 3$	56
5.4	An illustration of the vertex-splitting operation used for mapping G to G^{new}	61
5.5	Digraph from Figure 5.1 after the vertex-splitting operation	62
5.6	Transforming a dicycle C_j formed by an anti-parallel pair of thin edges in Figure 5.1 to C_j^{new} by the vertex-splitting operation.	62
5.7	$tour(e)$	62
6.1	An illustration of the role of the decomposition theorem for the Las system (Theorem 6.4.1) in our analysis.	76
6.2	Illustration of a stem s and its twin link ℓ	77
6.3	Illustration of the proof of Lemma 6.4.3.	83
6.4	Instance for $k = 6$	85

6.5	Worked example of the algorithm.	92
6.6	Illustration of deficient 3-leaf tree.	93
6.7	Illustration of auxiliary graph.	94
6.8	The links ℓ_v and ℓ_a in the proof of Theorem 6.7.3.	97
6.9	Addressing deficient 3-leaf trees by replacing M by M^{new}	98
6.10	Two building blocks of our example.	100
6.11	Instance for $k = 3$	100
7.1	Illustration of the argument flow related to the decomposition theorem by Las system.	103
7.2	Illustration of a bud b_0 in the subtree T_v	105
7.3	Illustration of $\mathcal{R}^{special}(b_0)$ for a bud b_0	106
7.4	Illustration of bad 2-stem tree.	115
7.5	Illustration of leafy 3-covers of T_v for Subcase 2.2.2 in the proof of Lemma 7.4.5.	119
7.6	Illustration of deficient 3-leaf tree.	126
7.7	Illustration of deficient 4-leaf trees.	127
7.8	Illustration of an M -covered bud in T	129
7.9	Illustration of auxiliary graph.	132
7.10	Illustration of a buddy link ℓ in the proof of Lemma 7.6.5	134
7.11	The links ℓ_u and ℓ_s in the proof of Lemma 7.6.5.	136
7.12	The links ℓ_v and ℓ_a in Subcase 1.2 of the proof of Theorem 7.6.6.	139
7.13	The links ℓ_v, ℓ_{a_2} and a_1b_1 in Subcase 2.1 of the proof of Theorem 7.6.6.	139

Chapter 1

Introduction

Combinatorial optimization is motivated by many real-life problems with widespread applications in transportation, scheduling, network design, etc. Essentially, combinatorial optimization is a branch of optimization that deals with the problems where we seek an optimal solution from a finite set of candidates. This finite set of candidates is formed by satisfying the constraints posed by the problem. Each candidate is called a feasible solution, and the set of all candidates is called the feasible set. For every optimization problem, there is an associated objective function that maps each feasible solution to a real value. The value of the objective function for a feasible solution is called the value of the solution. An optimal solution here means a feasible solution that minimizes (maximizes, respectively) the objective function if the input problem is a minimization (maximization, respectively) problem. Thus, the goal of a combinatorial optimization problem is to find an optimal solution with respect to a particular objective function from a finite set of feasible solutions. We mention that the problems considered in this thesis are all minimization problems and the objective functions are all nonnegative valued. Specifically, this thesis focuses on three important problems in combinatorial optimization, namely the Path Traveling Salesman Problem, the Asymmetric Traveling Salesman Problem, and the Tree Augmentation Problem.

Although the feasible set is finite, its cardinality increases exponentially with the input size of the instance. In fact, many combinatorial optimization problems are **NP-hard**. Thus, it is impossible to find an optimal solution in polynomial time unless $P=NP$. One alternative way to approach **NP-hard** combinatorial optimization problems is to search for a “near optimal” solution instead of an optimal one. This motivates the research on approximation algorithms. For a minimization problem, an α -approximation algorithm is a polynomial-time algorithm that always outputs a feasible solution whose value is guaranteed to be at most α times the value of an optimal solution ($\alpha > 1$). This factor α is called the approximation factor, and it measures the

accuracy of the approximation algorithm for solving the problem. One key focus of this thesis is on the design of approximation algorithms for **NP-hard** combinatorial optimization problems.

Many different approaches are used for the design of approximation algorithms in combinatorial optimization, and many approximation algorithms are specific to the problem. However, there is a fundamental technique commonly used in the design of approximation algorithms: linear programming (LP) relaxation. Most of the time, a combinatorial optimization problem can be formulated naturally as an integer program (IP) where the objective function and constraints are linear, and the variables are restricted to taking integral values. However, in general, solving an IP is still an **NP-hard** problem. The difficulty comes from the discrete structure of the feasible set of the IP. One idea is to relax the IP by allowing the variables to take fractional values such that the resulting linear program (LP) can be solved in polynomial time (we use the standard abbreviation LP to stand for linear programming or linear program). The resulting linear program is called an LP relaxation of the combinatorial optimization problem. However, a (fractional) optimal solution of the LP relaxation can be far from the optimal solution of the original optimization problem. To address this difficulty, one possible way is to strengthen the LP relaxation by adding more variables and constraints. The Lift-and-Project system provides a systematic tool to construct tightened relaxations.

For a relaxation of a combinatorial optimization problem, every feasible solution of the problem corresponds to an integral solution of the relaxation. The integral solution that optimizes the objective function of the relaxation is called an optimal integral solution. For an instance of a minimization problem, the integrality ratio of a relaxation is defined to be the ratio of the value of an optimal integral solution to the value of an optimal (fractional) solution of the relaxation. Furthermore, for a minimization problem, the integrality ratio of a relaxation is the worst-case (supremum) ratio over all instances. The integrality ratio measures the quality of a relaxation. Informally speaking, integrality ratios are closely related to approximation algorithms via “rounding”. Rounding a (fractional) optimal solution of a relaxation to an integral solution is a popular approach in the design of approximation algorithms. Consider a minimization problem. For all instances, if we could always round an (fractional) optimal solution of a relaxation to an integral solution such that the value of the integral solution is at most α times the value of the (fractional) optimal solution, then the integrality ratio of the relaxation would be at most α . Conversely, if we could establish a lower bound α for the integrality ratio of a relaxation, then for all instances, it is impossible to always round an optimal (fractional) solution to an integral solution within a factor lower than α . The other key focus of this thesis is on the integrality ratios of tightened relaxations generated by the Lift-and-Project systems.

The Traveling Salesman Problem (TSP) and its variants are celebrated **NP-hard** problems in the area of combinatorial optimization. Given a complete (undirected) graph with edge costs, the goal of TSP is to find a minimum-cost Hamiltonian cycle (a cycle that visits every vertex exactly

once). We say that the edge costs are metric if they satisfy the triangle inequalities. In this case, Christofides' algorithm [20] achieves an approximation factor of $\frac{3}{2}$. Approximation algorithms for TSP and its variants have been studied for over four decades. In this thesis, we study two important variants of TSP under metric costs: (metric) s - t path TSP and (metric) Asymmetric TSP (ATSP).

We first consider a critical special case of the metric s - t path TSP. Let G be a connected graph with unit cost on each edge. Let s, t be two given vertices in G . The goal of the s - t path graph-TSP is to find a minimum-cost Hamiltonian path from s to t in the metric completion of G . A result of Hoogeveen [39] gave a $\frac{5}{3}$ -approximation algorithm for this problem. An and Shmoys [2] provided a slightly improved performance guarantee of $(\frac{5}{3} - \epsilon)$. Then, Mömke and Svensson [53] gave a 1.586-approximation algorithm for the s - t path graph-TSP. Mucha [54] improved the analysis of [53] and obtained a $\frac{19}{12} + \epsilon \approx 1.58333$ approximation guarantee for any $\epsilon > 0$. Recently, Sebő and Vygen [62] gave the first $\frac{3}{2}$ -approximation algorithm by using ear decomposition and matroid intersection. In Chapter 3, we design a new LP-based $\frac{3}{2}$ -approximation algorithm for the s - t path graph-TSP. Compared with the algorithm from [62], our algorithm and its analysis are much simpler. It is known that the integrality ratio of the path version of the Held-Karp LP relaxation is lower bounded by $\frac{3}{2}$ even in this graphic metric case. Our algorithm implies that the integrality ratio of this LP relaxation is at most $\frac{3}{2}$ for the graphic metric. Hence, from this point of view, our algorithm obtains the best possible approximation guarantee achievable by an algorithm based on the Held-Karp LP relaxation for the s - t path graph-TSP. These results have been published in the single-authored paper [33].

Given a complete graph with metric edge costs and two fixed vertices s and t , the goal of the (metric) s - t path TSP is to find a minimum-cost Hamiltonian path between s and t . Hoogeveen's [39] $\frac{5}{3}$ -approximation guarantee had been the best one since 1991 until the paper [1] by An, Kleinberg, and Shmoys improved on the $\frac{5}{3}$ approximation guarantee and presented an algorithm that achieves an approximation guarantee of $\frac{1+\sqrt{5}}{2} \approx 1.61803$. Later, Sebő [61] further improved the approximation factor to $\frac{8}{5}$. Very recently, Vygen [67] obtained an approximation factor slightly better than $\frac{8}{5}$. In fact, the algorithm in [61] is the same as the algorithm in [1]. It is called randomized Christofides' algorithm. In more detail, we solve an LP relaxation to get an optimal solution x^* that can be written as a convex combination of spanning trees. We sample a spanning tree J from these spanning trees according to the probability distribution defined by the coefficients of the convex combination. Let T denote the set of vertices of J that have the wrong degree. Then, similarly to Christofides' algorithm, the algorithm adds the minimum-cost T -join to fix the wrong-degree vertices of J . The expected cost of the random solution is the sum of the expected cost of J , which is the cost of x^* , and the expected cost of the T -join. Any feasible solution of the T -join polyhedron provides a cost upper bound for the T -join. An et al. [1] introduced so-called correction vectors to construct a special type of fractional T -join.

The correction vectors were further analyzed in [61] to obtain a better approximation factor. In Chapter 4, we provide a simple correction vector to derive the results of both [1] and [61]. An et al. [1] and Sebő [61] use two different LP relaxations of the s - t path TSP in their algorithms. [1] uses the path version of the Held-Karp LP relaxation for TSP, whereas [61] uses a weaker LP relaxation. This motivates a comparison of these two LP relaxations in Chapter 4. Finally, an instance that presents obstructions to two natural methods that aim for an approximation factor of $\frac{3}{2}$ is also discussed in Chapter 4. These results appear in the single-authored preprint [34], which has been accepted for publication in SIAM Journal on Discrete Mathematics.

LP relaxations give a powerful technique to design approximation algorithms for combinatorial optimization problems. All of our discussions on the s - t path TSP are based on LP relaxations. An intriguing possibility for obtaining better approximation factors would be to tighten the LP relaxations. Some Lift-and-Project systems have been developed in order to obtain tightenings of relaxations in a systematic manner.

Assume that each variable in the initial LP relaxation is in the interval $[0, 1]$, i.e., the integral solutions are zero/one, and let n denote the number of variables in the LP relaxation. A Lift-and-Project system starts with the LP relaxation, and then iteratively obtains a sequence of stronger relaxations such that the associated feasible regions form a nested family that contains (and converges to) the integral hull. By the integral hull we mean the convex hull of the zero-one solutions that are feasible for the original relaxation. The index of each relaxation in the sequence of tightened relaxations is known as the level of the system; the level of the original relaxation is defined to be zero, and the relaxation at level n is exact, i.e., the associated feasible region is equal to the integral hull. In particular, Sherali and Adams [63] devised the Sherali-Adams (SA) system, Lovász and Schrijver [50] devised the Lovász-Schrijver (LS) system, and Lasserre [46] devised the Lasserre (Las) system. See [47] for a survey of these systems; several other Lift-and-Project systems are known, see [19, 6].

This thesis focuses on the Sherali-Adams system and the Lasserre system. These two Lift-and-Project systems strengthen relaxations in a “global manner”. On the one hand, this enhances its algorithmic leverage for deriving positive results. On the other hand, it also makes it more challenging to design instances with bad integrality ratios for the sequence of relaxations derived by these Lift-and-Project systems. Over the last two decades, a number of important improvements on approximation guarantees have been achieved based on relaxations obtained from Lift-and-Project systems. See [19] for a survey of many such positive results. Meanwhile, starting with the work of Stephen and Tunçel [64] and Arora et al. [4], substantial research efforts have also been devoted to showing that tightened relaxations (for many levels) fail to reduce the integrality ratios for some combinatorial optimization problems; also, see [19] for a list of negative results. In this thesis, we show negative results and positive results for the Sherali-Adams system and the Lasserre system, respectively.

Given a complete directed graph with metric edge costs, the goal of ATSP is to find a minimum-cost directed Hamiltonian cycle. Many LP relaxations are known for ATSP, see [57] for a recent survey. The best known is due to Dantzig, Fulkerson and Johnson; we call it the standard LP relaxation or the DFJ LP relaxation. To the best of our knowledge, there are two previous papers with results on the integrality ratio for Lift-and-Project systems applied to the TSP and its variants. Cheung [18] proved an integrality ratio of $\frac{4}{3}$ for the TSP, for $O(1)$ levels of LS_+ , the semidefinite programming (SDP) version of the Lovász-Schrijver system [50]. For ATSP, there is a further LP relaxation of the standard LP relaxation, called the balanced LP relaxation. Watson [68] proved an integrality ratio of $\frac{3}{2}$ for level 1 of the Lovász-Schrijver system starting with the balanced LP relaxation for ATSP (in fact, both the systems LS and SA give the same relaxation at level one). In Chapter 5, we consider the SA system starting with two LP relaxations of ATSP, and we prove lower bounds on the integrality ratios that survive for multiple levels; the two relaxations are the standard LP and the balanced LP. These results have been published in the joint-authored paper [14].

Given a connected (undirected) graph G with nonnegative costs (weights) on the edges, together with a spanning tree T of G , the goal of the weighted Tree Augmentation Problem is to find a set of edges, $F \subseteq E(G) - E(T)$, of minimum cost such that the graph $(V, E(T) \cup F)$ is 2-edge connected. Frederickson and Jaja [32] in 1981 presented a 2-approximation algorithm for the weighted Tree Augmentation problem. In 2001, Jain [40] invented a 2-approximation iterative rounding algorithm for a more general problem, namely the Survivable Network Design Problem. To date, the best known approximation factor for the weighted Tree Augmentation Problem is 2. When the edge costs are uniform, we get the unweighted Tree Augmentation Problem. It has been proved that even the unweighted Tree Augmentation Problem is APX-hard, see [43, Section 4]. Since our focus is on the unweighted version, we use the abbreviation TAP for the *unweighted* Tree Augmentation Problem. There have been some important advances for TAP. Nagamochi [55] first beat the 2-approximation factor and presented a $(1.875 + \epsilon)$ -approximation algorithm for TAP. Subsequently, Even et al. [29] built on the ideas and techniques initiated by Nagamochi and presented an elegant algorithm and analysis that achieves an approximation guarantee of 1.8. In a conference publication from 2001, Even et al. [28] reported the first 1.5-approximation algorithm for TAP. Very recently, Kortsarz and Nutov finalized the journal version of this result [44].

For TAP, there is a natural LP relaxation, called the covering LP relaxation. A lower bound of 1.5 on the integrality ratio is known [17]. However, the best known upper bound for the integrality ratio of the covering LP relaxation is 2, and this is implied by Jain's result [40]. Whether the integrality ratio of the covering LP relaxation is 1.5 or worse than 1.5 is unknown. A related open question is whether there exists either an LP relaxation or an SDP relaxation with integrality ratio at most 1.5. In Chapters 6 and 7, we present an SDP relaxation of TAP with

integrality ratio at most $1.5 + \epsilon$, where ϵ can be any small positive constant. This SDP relaxation is obtained by applying t levels of the Lasserre system to an LP relaxation of TAP where t depends on ϵ . A $(1.5 + \epsilon)$ -approximation algorithm is also given for TAP. The algorithm follows the algorithmic scheme of [29]. Our analysis of the integrality ratio of the SDP relaxation is based on this algorithm. Also, our analysis relies on the decomposition theorem for the Lasserre system due to Karlin and Mathieu and Nguyen [41]. These results are based on the joint-authored papers [12] [13].

This thesis is organized as follows. Chapter 2 gives some notation and presents LP relaxations for the optimization problems considered in this thesis. Chapter 2 also has a very brief introduction to Lift-and-Project systems. Chapter 3 presents an LP-based 1.5-approximation algorithm for the s - t path graph-TSP. In Chapter 4, we study the metric s - t path TSP. Starting from Chapter 5, we turn our attention to results in the framework of Lift-and-Project systems. Chapter 5 shows lower bounds for the integrality ratios of the Sherali-Adams system applied to two LP relaxations of ATSP. Chapters 6 and 7 focus on TAP via the Lasserre system. Chapter 6 is for a special case and Chapter 7 has general results which subsume Chapter 6 but are substantially more difficult. Chapter 6 serves as an overview of Chapter 7. Finally, Chapter 8 concludes the thesis.

Chapter 2

Preliminaries

In this chapter, we give some notation and present the LP relaxations for the problems considered in the thesis. A very brief introduction to Lift-and-Project systems is given as well.

2.1 LP relaxations

Let $G = (V, E)$ be a (undirected) graph. We say G is *connected* if for any two vertices u, v in V , there is a path between u and v in G . Furthermore, if for any two vertices u, v in V , there are at least two edge-disjoint paths between u and v in G , then G is said to be *2 edge-connected*.

We call a nonempty, proper subset of vertices S a *cut*. For a cut S , we define $\delta_G(S) = \{(u, v) \in E : u \in S, v \notin S\}$. If $S = \{v\}$, then we use $\delta_G(v)$ instead of $\delta_G(\{v\})$. Furthermore, if there is no ambiguity, we use $\delta(S)$ for short. For any vertex v , the cardinality of $\delta(v)$ is called *degree of v* .

For a partition $\mathcal{W} = \{W_1, W_2, \dots, W_\ell\}$ of the vertex set V , let $\delta(\mathcal{W})$ denote $\cup_{1 \leq i \leq \ell} \delta(W_i)$. Let s, t be two vertices in V . For a cut S , if $|S \cap \{s, t\}| = 1$, then S is called an *s - t cut*. For a subset S of V , we let $E(S)$ denote the set of edges induced by S , thus, $E(S) = \{(u, v) \in E : u, v \in S\}$.

For a vector $x \in \mathbb{R}^A$, we define $x(D) = \sum_{e \in D} x_e$ for any subset D of A . When $D = \{e\}$, we may use x_e or $x(e)$. For any two sets A and B , we use $A - B$ to denote $\{e \in A : e \notin B\}$. For simplicity, we may denote the addition (removal) of a single item e to (from) a set A by $A + e$ ($A - e$, respectively), rather than by $A \cup \{e\}$ ($A - \{e\}$, respectively).

Given a complete graph $G = (V, E)$ with metric edge costs c , the goal of (metric) TSP is to *find a minimum-cost Hamiltonian cycle (a cycle that visits every vertex exactly once)*. We say that the edge costs c are *metric* if c satisfies the triangle inequalities, i.e., $c((u, v)) + c((v, w)) \geq c((u, w))$ for any three vertices $u, v, w \in V$. The following well known linear program is a relaxation of TSP on G . This LP relaxation is first introduced by Dantzig, Fulkerson and Johnson [23] but it is also called Held-Karp LP relaxation [37], [38]; also see the survey paper of Vygen [66].

$$\begin{aligned}
\text{(DFJ LP/Held-Karp LP)} \quad & \text{minimize : } \sum_{e \in E} c_e x_e \\
& \text{subject to : } x(\delta(v)) = 2 & \forall v \in V \\
& x(\delta(S)) \geq 2 & \forall \emptyset \subsetneq S \subsetneq V \\
& 1 \geq x_e \geq 0 & \forall e \in E
\end{aligned}$$

We consider two versions of this LP relaxation for the s - t path TSP and ATSP, respectively. Following the literature, the analogous LP relaxation for the s - t path TSP is called *path Held-Karp LP relaxation*, whereas the analogous LP relaxation for ATSP is called *DFJ LP relaxation*.

2.1.1 Path TSP

Given a complete graph $G = (V, E)$ with metric edge costs c and two fixed vertices s, t , the goal of (metric) s - t path TSP is to *find a minimum-cost Hamiltonian path from s to t (a path from s to t that visits every vertex exactly once)*. The path Held-Karp LP relaxation for the (metric) s - t path TSP is defined as follows:

$$\begin{aligned}
\text{(Path Held-Karp LP)} \quad & \text{minimize : } \sum_{e \in E} c_e x_e \\
& \text{subject to : } x(\delta(s)) = x(\delta(t)) = 1 \\
& x(\delta(v)) = 2 & \forall v \neq s, t \\
& x(\delta(S)) \geq 1 & \forall s\text{-}t \text{ cut } S \\
& x(\delta(S)) \geq 2 & \forall \emptyset \subsetneq S \subsetneq V, |S \cap \{s, t\}| \text{ even} \\
& 1 \geq x_e \geq 0 & \forall e \in E
\end{aligned}$$

The (metric) s - t path TSP is defined on a complete graph with metric edge costs. There is an equivalent definition of the s - t path TSP starting with a connected graph $H = (V, E(H))$ with nonnegative edge costs c^H . Let s, t be two fixed vertices. Let $G = (V, E)$ be the metric completion of H with metric edge costs c . By the *metric completion* of a connected graph H , we mean the complete graph on V with the edge costs, where the cost of an edge between v, w is taken to be the minimum cost (w.r.t. c^H) of a path between v, w on H . Let $2H$ be the graph obtained from H by doubling every edge of H . The s - t path TSP on G is equivalent to the problem of finding a minimum-cost trail in $2H$ from s to t visiting every vertex at least once (multiple visits are allowed for the vertices but not the edges). Thus, the problem is to *find a minimum-cost connected spanning subgraph of $2H$ with $\{s, t\}$ as the odd-degree vertex set*. Hence, the following linear program with constraints on all partitions is another LP relaxation of the s - t path TSP.

$$\begin{aligned}
(\text{Partition LP}) \quad & \text{minimize : } \sum_{e \in E(H)} c_e^H x_e \\
& \text{subject to : } x(\delta(\mathcal{W})) \geq |\mathcal{W}| - 1 \quad \forall \text{ partition } \mathcal{W} \text{ of } V \\
& \quad \quad \quad x(\delta(S)) \geq 2 \quad \forall \emptyset \subsetneq S \subsetneq V, |S \cap \{s, t\}| \text{ even} \\
& \quad \quad \quad x_e \geq 0 \quad \forall e \in E(H)
\end{aligned}$$

Note that the Partition LP relaxation is defined on the original graph H but the path Held-Karp LP relaxation is defined on the metric completion G of H .

For the s - t path TSP, there is a closely related concept, called T -join. Consider a connected graph $K = (V, E(K))$ with nonnegative edge costs c^K . Let T be a nonempty subset of V with $|T|$ even. For $F \subseteq E(K)$, if the set of odd degree vertices of the graph (V, F) is T , then we call F a T -join. Since K is connected, a T -join always exists. For any $\emptyset \subsetneq S \subsetneq V$, if $|S \cap T|$ is odd (even, respectively), then we call S a T -odd cut (T -even cut, respectively). The following LP formulates the problem of finding a T -join of minimum cost:

$$\begin{aligned}
(T\text{-join Polyhedron}) \quad & \text{minimize : } \sum_{e \in E(K)} c_e^K x_e \\
& \text{subject to : } x(\delta(S)) \geq 1 \quad \forall T\text{-odd } S \\
& \quad \quad \quad x_e \geq 0 \quad \forall e \in E(K)
\end{aligned}$$

Lemma 2.1.1 [26] *Let $K = (V, E(K))$ be a connected graph with nonnegative edge costs c^K*

and T be a nonempty subset of V with even size. Then, the optimal value of the T -join polyhedron is the same as the minimum cost of a T -join on K .

2.1.2 Asymmetric TSP

Let $G = (V, E)$ be a directed graph (digraph). We say G is *strongly connected* if for any two vertices u, v , there is a directed path from u to v in G . If replacing all directed edges in G by undirected edges results in a connected (undirected) graph, then G is said to be *weakly connected*.

For a vertex subset $\emptyset \subsetneq S \subsetneq V$, $\delta_G^{out}(S)$ denotes $\{(v, w) \in E : v \in S, w \notin S\}$, and $\delta_G^{in}(S)$ denotes $\{(v, w) \in E : v \notin S, w \in S\}$. Similarly, if $S = \{v\}$, then we use $\delta_G^{out}(v)$ instead of $\delta_G^{out}(\{v\})$ and $\delta_G^{in}(v)$ instead of $\delta_G^{in}(\{v\})$. Furthermore, if there is no ambiguity, we use $\delta^{out}(S)$ and $\delta^{in}(S)$ for short. For a vertex v , the cardinality of $\delta^{in}(v)$ is called *indegree of v* , and the cardinality of $\delta^{out}(v)$ is called *outdegree of v* .

For ATSP, the input graph $G = (V, E)$ is a complete digraph with metric edge costs c . The goal of ATSP is to find a *minimum-cost directed Hamiltonian cycle* (a directed cycle that visits every vertex exactly once). On the complete digraph G , we say that the edge costs c are *metric* if c satisfies the triangle inequalities for the directed edges, i.e., $c((u, v)) + c((v, w)) \geq c((u, w))$ for any three vertices $u, v, w \in V$ where (a, b) means a directed edge from a to b .

The following is the well known version of the DFJ LP relaxation for ATSP; this LP is also called *standard LP relaxation* of ATSP.

$$\begin{aligned}
\text{(DFJ LP/Standard LP)} \quad & \text{minimize: } \sum_{e \in E} c_e x_e \\
& \text{subject to: } x(\delta^{in}(S)) \geq 1 & \forall \emptyset \subsetneq S \subsetneq V \\
& x(\delta^{out}(S)) \geq 1 & \forall \emptyset \subsetneq S \subsetneq V \\
& x(\delta^{in}(v)) = 1, x(\delta^{out}(v)) = 1 & \forall v \in V \\
& 0 \leq x_e \leq 1 & \forall e \in E
\end{aligned}$$

For ATSP, there is a further LP relaxation of the DFJ LP relaxation that is obtained by replacing the indegree and outdegree constraints for each vertex by the balanced degree constraint $x(\delta^{in}(v)) = x(\delta^{out}(v))$; this LP is called the *balanced LP relaxation*.

$$\begin{aligned}
\textbf{(Balanced LP)} \quad & \text{minimize: } \sum_{e \in E} c_e x_e \\
& \text{subject to: } x(\delta^{in}(S)) \geq 1 & \forall \emptyset \subsetneq S \subsetneq V \\
& x(\delta^{out}(S)) \geq 1 & \forall \emptyset \subsetneq S \subsetneq V \\
& x(\delta^{in}(v)) = x(\delta^{out}(v)) & \forall v \in V \\
& 0 \leq x_e \leq 1 & \forall e \in E
\end{aligned}$$

2.1.3 TAP

For TAP, the input is a connected (undirected) graph $G = (V, E(G))$ with a spanning tree $T = (V, \hat{E}_T)$ of G . Let $E = E(G) - \hat{E}_T$. An edge in \hat{E}_T is called a *tree-edge*, whereas an edge in E is called a *link*. The goal of TAP is to *find a minimum-size subset F of E such that the graph $(V, \hat{E}_T \cup F)$ is 2-edge connected.*

We say that a link uv *covers* a tree-edge \hat{e} if \hat{e} is on the unique path of the tree T between u and v . For any tree-edge $\hat{e} \in \hat{E}_T$, we use $\delta_E(\hat{e})$ to denote the set of links that cover \hat{e} . Finding a subset F of E such that $(V, \hat{E}_T \cup F)$ is 2-edge connected is equivalent to *finding a subset F of E such that every tree-edge is covered by some link in F .* The following is a natural LP relaxation of TAP, called the *covering LP relaxation*.

$$\begin{aligned}
\textbf{(Covering LP)} \quad & \text{minimize : } \sum_{uv \in E} x_{uv} \\
& \text{subject to : } \sum_{uv \in \delta_E(\hat{e})} x_{uv} \geq 1 & \forall \hat{e} \in \hat{E}_T \\
& 0 \leq x_{uv} \leq 1 & \forall uv \in E
\end{aligned}$$

2.2 Lift-and-Project systems

A Lift-and-Project system is a systematic method to tighten a given LP relaxation iteratively to finally converge to the integral hull. In this section, we show three popular Lift-and-Project systems: Lovász-Schrijver system [50], Sherali-Adams system [63], and Lasserre system [46]. Although all these systems provide the tightened relaxations for the initial LP relaxation, the Lovász-Schrijver and Sherali-Adams systems generate linear programs, whereas the Lasserre

system generates semidefinite programs. The Lovász-Schrijver system has a stronger SDP version also shown in this section. In the following, we first give the formal definitions of these Lift-and-Project systems. But we mention that the main results in this thesis can be stated and proved without going into these formalities. The properties presented after the definitions for these Lift-and-Project systems are more critical to our discussion in this thesis. We give the definitions for the sake of completeness.

Let $\hat{P} := \{y \in [0, 1]^n : g_l(y) \geq 0 \text{ for } l = 1, 2, \dots, m\}$ be a polytope where every $g_l(y) \geq 0$ is a linear constraint. Let $\text{cov}(\hat{P} \cap [0, 1]^n)$ be the convex hull generated by the integral solutions of \hat{P} . We call $\text{cov}(\hat{P} \cap [0, 1]^n)$ the *integral hull* of \hat{P} . For an LP relaxation \hat{P} of a combinatorial optimization problem, optimizing a linear function over the integral hull of \hat{P} is always equivalent to the problem itself. Hence, the goal is to tighten \hat{P} to approach the integral hull of \hat{P} as close as possible. The Lift-and-Project system provides a systematic tool for this purpose.

Let $P := \{\lambda \begin{pmatrix} 1 \\ y \end{pmatrix} : \lambda \geq 0, y \in \hat{P}\}$ be the associated cone with \hat{P} . When we talk about a general polytope and its associated cone for the Lift-and-Project systems, we use an accented symbol to denote a polytope, e.g., \hat{P} , and the symbol (without accent) to denote the associated cone, e.g., P . This simplifies the notation in Chapter 5. We define $\mathcal{M}(P)$ to be the set of symmetric matrices $Y = (Y_{ij}) \in \mathbb{R}^{(n+1) \times (n+1)}$ whose rows and columns are indexed by $\{0, 1, \dots, n\}$ satisfying: (i) $Y_{i,i} = Y_{0,i}$ for $1 \leq i \leq n$; (ii) $\text{col}_i(Y), \text{col}_0(Y) - \text{col}_i(Y) \in P$ for $1 \leq i \leq n$ where $\text{col}_i(Y)$ is the i th column of Y . Let $\text{LS}(\hat{P}) := \{y \in \mathbb{R}^n : \begin{pmatrix} 1 \\ y \end{pmatrix} = \text{col}_0(Y) \text{ for some } Y \in \mathcal{M}(P)\}$. For any positive integer t , we define $\text{LS}^t(\hat{P})$ iteratively by $\text{LS}^t(\hat{P}) = \text{LS}(\text{LS}^{t-1}(\hat{P}))$. Then $\text{LS}^t(\hat{P})$ is called *level t of the Lovász-Schrijver system starting with \hat{P}* .

A symmetric $d \times d$ matrix A is said to be *positive semidefinite* if $x^T A x \geq 0, \forall x \in \mathbb{R}^d$; $A \succeq 0$ denotes that A is symmetric and positive semidefinite. There is a SDP version of Lovász-Schrijver system requiring the matrix Y to be positive semidefinite. That is, we let $\text{LS}_+(\hat{P}) := \{y \in \mathbb{R}^n : \begin{pmatrix} 1 \\ y \end{pmatrix} = \text{col}_0(Y) \text{ for some } Y \in \mathcal{M}(P) \text{ such that } Y \succeq 0\}$. Similarly, $\text{LS}_+^t(\hat{P})$ is defined iteratively by $\text{LS}_+^t(\hat{P}) = \text{LS}_+(\text{LS}_+^{t-1}(\hat{P}))$. Then $\text{LS}_+^t(\hat{P})$ is called *level t of the SDP version of Lovász-Schrijver system starting with \hat{P}* .

Let N denote $\{1, 2, \dots, n\}$. Let $\mathcal{P}(N)$ be the collection of all subsets of N . For $1 \leq t \leq n$, we denote by $\mathcal{P}_t(N)$ the family of all subsets of N of size at most t ; thus $\mathcal{P}_t(N) = \{S \subseteq N : |S| \leq t\}$. We may abbreviate $\mathcal{P}(N)$ to \mathcal{P} and $\mathcal{P}_t(N)$ to \mathcal{P}_t .

Let $v \in \mathbb{R}^{\mathcal{P}(N)}$. We define $M(v) \in \mathbb{R}^{\mathcal{P}(N) \times \mathcal{P}(N)}$ to be the matrix whose (I, J) -entry is $v_{I \cup J}$. For any $U \subseteq N$, we denote by $M_U(v)$ the submatrix of $M(v)$ indexed by all subsets of

U . Let $M_t(v)$ denote the submatrix of $M(v)$ that is indexed by all sets $I \in \mathcal{P}_t(N)$. For two vectors $u, v \in \mathbb{R}^{\mathcal{P}(N)}$, we define $u * v = M(v)u$. Note that the constraints of \hat{P} are denoted by $g_l(y) \geq 0, \forall l \in \{1, \dots, m\}$. The constraint can be viewed as a vector in $\mathbb{R}^{\mathcal{P}(N)}$. Specifically, suppose $g_l(y) \geq 0$ is of the form $\sum_{i=1}^n a_i y_i - b \geq 0$. Then, the constraint can be viewed as a vector g_l in $\mathbb{R}^{\mathcal{P}(N)}$ with $(g_l)_\emptyset = -b$, $(g_l)_{\{i\}} = a_i$ for $i \in N$, $(g_l)_I = 0$ for $|I| \geq 2$. For $1 \leq t \leq n$, we define

$$\text{SA}^t(P) := \{y \in \mathbb{R}^{\mathcal{P}_{t+1}} : M_U(g_l * y) \succeq 0 \text{ for all } U \in \mathcal{P}_t \text{ and } \forall l \in \{1, \dots, m\} \\ M_W(y) \succeq 0 \text{ for all } W \in \mathcal{P}_{t+1}\},$$

$$\text{Las}^t(P) := \{y \in \mathbb{R}^{\mathcal{P}_{2t+2}} : M_t(g_l * y) \succeq 0, M_{t+1}(y) \succeq 0 \text{ for } \forall l \in \{1, \dots, m\}\}.$$

Although the “*” operator requires y to be defined on $\mathbb{R}^{\mathcal{P}(N)}$, we can tell that M_U , M_W , and M_{t+1} are only concerned with the coordinates of y on a subset of $\mathcal{P}(N)$. That is, $\text{SA}^t(P)$ and $\text{Las}^t(P)$ only require $y \in \mathbb{R}^{\mathcal{P}_{t+1}}$ and $y \in \mathbb{R}^{\mathcal{P}_{2t+2}}$, respectively. Another way to address the problem of definition domain here is to extend the vector $y \in \mathbb{R}^{\mathcal{P}_{t+1}}$ or $y \in \mathbb{R}^{\mathcal{P}_{2t+2}}$ to a vector in $\mathbb{R}^{\mathcal{P}(N)}$ by setting every undefined coordinate to be zero.

Let $\text{SA}^t(\hat{P}) := \{y \in \mathbb{R}^{\mathcal{P}_{t+1}} : y_\emptyset = 1, y \in \text{SA}^t(P)\}$ and $\text{Las}^t(\hat{P}) := \{y \in \mathbb{R}^{\mathcal{P}_{2t+2}} : y_\emptyset = 1, y \in \text{Las}^t(P)\}$. Then, $\text{SA}^t(\hat{P})$ is called *level t of the Sherali-Adams system starting with \hat{P}* , and $\text{Las}^t(\hat{P})$ is called *level t of the Lasserre system starting with \hat{P}* . For the notation for **SA** and **Las** systems, we use the convention that a Lift-and-Project system applying to a cone (polytope, respectively) results in a cone (polytope, respectively). For example, $\text{SA}^t(P)$ is a cone, whereas $\text{SA}^t(\hat{P})$ is a polytope.

Note that $\text{SA}^t(\hat{P})$ and $\text{Las}^t(\hat{P})$ have higher dimensions than the initial polytope \hat{P} . Let $\text{SA}_{proj}^t(\hat{P})$ ($\text{Las}_{proj}^t(\hat{P})$, respectively) be the projection of $\text{SA}^t(\hat{P})$ ($\text{Las}^t(\hat{P})$, respectively) on the subspace \mathbb{R}^n indexed by the singleton sets. We mention that the higher dimensional set $\text{SA}^t(\hat{P})$ ($\text{Las}^t(\hat{P})$, respectively) and its projected set $\text{SA}_{proj}^t(\hat{P})$ ($\text{Las}_{proj}^t(\hat{P})$, respectively) have no difference in terms of the integrality ratios. This is due to the fact that the cost function is only defined on the variables in the initial LP relaxation, i.e., the variables indexed by the singleton sets. Thus, for the integrality ratios, we can consider either $\text{SA}^t(\hat{P})$ ($\text{Las}^t(\hat{P})$, respectively) or $\text{SA}_{proj}^t(\hat{P})$ ($\text{Las}_{proj}^t(\hat{P})$, respectively).

We can tell from the definitions that $\text{Las}_{proj}^t(\hat{P})$ is stronger than $\text{SA}_{proj}^t(\hat{P})$. A comparison of these Lift-and-Project systems given above is shown in Figure 2.1. We mention that the Sherali-Adams system is incomparable with the SDP version of the Lovász-Schrijver system.

Although the Sherali-Adams system can be defined by positive semidefiniteness of the relevant matrices as above, there is an equivalent definition only using linear systems (see [47] for

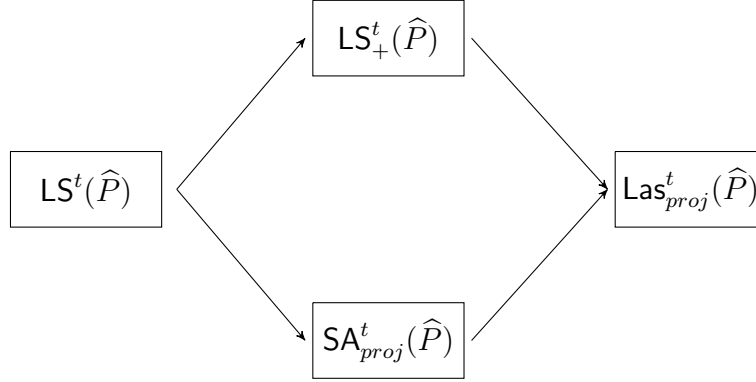


Figure 2.1: A comparison of the Lift-and-Project systems. $A \rightarrow B$: B is stronger than A

a proof of the equivalence).

Linearized Sherali-Adams system: Let the linear constraints of \hat{P} be of the form $\sum_{i=1}^n a_i y_i \geq b$. Here we include the constraints $y_i \geq 0$ and $-y_i \geq -1$ for all $1 \leq i \leq n$ into the system. Then, $SA^t(\hat{P})$ is a linear program over the variables $\{y_S : S \subseteq \{1, 2, \dots, n\}, |S| \leq t+1\}$ (thus, $y \in \mathbb{R}^{\mathcal{P}_{t+1}}$) with the constraints $y_\emptyset = 1$ and

$$\sum_{i=1}^n a_i \sum_{\emptyset \subseteq T \subseteq Q} (-1)^{|T|} y_{S \cup T \cup \{i\}} \geq b \sum_{\emptyset \subseteq T \subseteq Q} (-1)^{|T|} y_{S \cup T}.$$

for every original constraint $\sum_{i=1}^n a_i y_i \geq b$ and for every disjoint $S, Q \subseteq \{1, \dots, n\}$ with $|S| + |Q| \leq t$.

A Lift-and-Project system generates a sequence of nested relaxations that converges to the integral hull of the initial LP relaxation \hat{P} . In particular, the level n of the Lift-and-Project system is the same as the integral hull of \hat{P} . This can be stated as follows:

$$\hat{P} = Op^0(\hat{P}) \supseteq Op^1(\hat{P}) \supseteq Op^2(\hat{P}) \supseteq \dots \supseteq Op^n(\hat{P}) = \text{conv}(\hat{P} \cap [0, 1]^n)$$

where Op can be $LS, LS_+, SA_{proj}, Las_{proj}$. The level of the original relaxation is defined to be zero.

For the Sherali-Adams system and for any fixed constant t , it is known that the LP relaxation at level t of the Sherali-Adams system can be solved to optimality in polynomial time, assuming that the original relaxation has a polynomial-time separation oracle [65]. For the Lasserre system applied to a polynomial-time size LP and for any fixed constant t , the SDP relaxation

at level t of the **Lasserre** system is of polynomial-time size as well, and thus it can be solved to optimality (up to a “small enough” additive error term) in polynomial time.

Chapter 3

Path Graph Traveling Salesman Problem

In this chapter,¹ we present a new 1.5-approximation algorithm for the s - t path graph-TSP. A result of Hoogeveen [39] gave a $\frac{5}{3}$ -approximation algorithm for this problem. An and Shmoys [2] provided a slightly improved performance guarantee of $(\frac{5}{3} - \epsilon)$. Then, Mömke and Svensson [53] gave a 1.586-approximation algorithm for the s - t path graph-TSP. Mucha [54] improved the analysis of [53] and obtained a $\frac{19}{12} + \epsilon \approx 1.58333$ approximation guarantee for any $\epsilon > 0$. Recently, Sebő and Vygen [62] gave the first 1.5-approximation algorithm by using ear decomposition and matroid intersection.

Compared with the algorithm from Sebő and Vygen [62], our algorithm is conceptually simpler and its analysis is much shorter. The key point of our algorithm is to *find a minimum spanning tree that intersects every narrow cut in an odd number of edges*. Such a tree guarantees that the number of edges fixing the wrong degree vertices is at most half of the optimal value of the LP relaxation. Finally, the union of the spanning tree and the added edges provide us the 1.5-approximation guarantee. The detailed description of our algorithm is presented in Section 3.2. The graphic property is used to guarantee that the cost of the spanning tree we find is bounded by the optimum of the LP relaxation. However, for the general metric case, there exists an example such that the cost of the spanning tree in our algorithm is strictly larger than the optimum of the LP relaxation (see Section 4.4).

¹ The results of this chapter have already been published [33].

3.1 Preliminaries

Let $H = (V, E(H))$ be a connected graph with unit cost c^H on each edge in $E(H)$. Let s, t be two given vertices in H . Let G be the metric completion of H with metric edge costs c . The goal of the s - t path graph-TSP is to find a *minimum-cost Hamiltonian path from s to t on G w.r.t. the edge costs c* . Denote the cost of this path by $PTSP_{opt}(H)$. Recall from Section 2.1.1 that the s - t path graph-TSP on G is equivalent to finding a minimum-size connected spanning subgraph of $2H$ with $\{s, t\}$ as the odd-degree vertex set. The Partition LP defined on H in Section 2.1.1 is an LP relaxation of the s - t path graph-TSP. Note that $c_e^H = 1$ for any $e \in E(H)$ in this case. We restate the Partition LP for unit edge costs as follows:

$$\begin{aligned}
 & \text{minimize : } \sum_{e \in E(H)} x_e & \text{(LP1)} \\
 & \text{subject to : } x(\delta(\mathcal{W})) \geq |\mathcal{W}| - 1 & \forall \text{ partition } \mathcal{W} \text{ of } V \\
 & & x(\delta(S)) \geq 2 & \forall \emptyset \subsetneq S \subsetneq V, |S \cap \{s, t\}| \text{ even} \\
 & & x_e \geq 0 & \forall e \in E(H)
 \end{aligned}$$

This LP relaxation is defined on the original graph. In the graphic case, every spanning tree in the original graph has minimum cost. We will use this fact to bound the cost of the spanning tree in our algorithm. Let x^* be an optimal solution of LP1. Note that LP1 can be solved in polynomial time via the ellipsoid method [36]. We know that $\sum_{e \in E(H)} x_e^* \leq PTSP_{opt}(H)$.

Let Q be an s - t cut. If $x^*(\delta(Q)) < 2$, we call it a *narrow cut* (for the solution x^* of LP1).

Lemma 3.1.1 [1, Lemma 1] *Let $Q_1, Q_2 \subseteq V$ be two distinct narrow cuts such that $s \in Q_1$ and $s \in Q_2$ (for the solution x^* of LP1). Then $Q_1 \subsetneq Q_2$ or $Q_2 \subsetneq Q_1$.*

Proof. Suppose that the statement is false. Then both $Q_1 - Q_2$ and $Q_2 - Q_1$ are nonempty. Note that both $Q_1 - Q_2$ and $Q_2 - Q_1$ are $\{s, t\}$ -even. Hence, $x^*(\delta(Q_1)) + x^*(\delta(Q_2)) \geq x^*(\delta(Q_1 - Q_2)) + x^*(\delta(Q_2 - Q_1)) \geq 4$ by the constraints in LP1. However, $x^*(\delta(Q_1)) + x^*(\delta(Q_2)) < 4$. This is a contradiction. \square

Hence, we know that the set of narrow cuts containing s forms a nested family. Let Q_1, Q_2, \dots, Q_k be all the narrow cuts containing s such that $s \in Q_1 \subsetneq Q_2 \subsetneq Q_3 \cdots \subsetneq Q_k \subsetneq V$. Define $L_i = Q_i - Q_{i-1}$ for $i = 1, 2, \dots, k, k+1$ where $Q_0 = \phi$ and $Q_{k+1} = V$. Note that each L_i is nonempty and $\cup_{1 \leq i \leq k+1} L_i = V$.

In the rest of this section, we address properties of the edge set needed for fixing the wrong degree vertices of the spanning tree computed by the algorithm. Let T be a nonempty subset of V with $|T|$ even. Recall from Section 2.1.1 that the following LP is the T -join polyhedron that formulates the problem of finding a T -join of minimum size on H :

$$\begin{aligned}
& \text{minimize : } \sum_{e \in E(H)} x_e & (\text{LP2}) \\
& \text{subject to : } x(\delta(S)) \geq 1 & \forall T\text{-odd } S \\
& x_e \geq 0 & \forall e \in E(H)
\end{aligned}$$

Let K be a spanning tree with vertex set V . When there is no risk of confusion, we will use the same notation K for the spanning tree K and its edge set $E(K)$. The set of wrong degree vertices of K is defined as $\{v \in \{s, t\} : |\delta(v) \cap K| \text{ even}\} \cup \{v \in V - \{s, t\} : |\delta(v) \cap K| \text{ odd}\}$.

Lemma 3.1.2 [1] *Let T be the set of wrong degree vertices of a spanning tree K . Let S be an s - t cut. If S is T -odd, then $|\delta(S) \cap K|$ is even.*

Proof. Since $\sum_{v \in S} |\delta(v) \cap K| = 2|E(S) \cap K| + |\delta(S) \cap K|$, we have $|\delta(S) \cap K|$ has the same parity as $\sum_{v \in S} |\delta(v) \cap K|$. Without loss of the generality, we assume $s \in S, t \notin S$. By the definition of T , we know that $(S - \{s\}) \cap T$ is the set of vertices v in $S - \{s\}$ such that $|\delta(v) \cap K|$ is odd. If $|\delta(s) \cap K|$ is odd, then $s \notin T$. In this case, since S is T -odd, $|(S - \{s\}) \cap T|$ is odd. Hence, we have an even number of vertices v in S such that $|\delta(v) \cap K|$ is odd, which implies that $\sum_{v \in S} |\delta(v) \cap K|$ is even. Otherwise, $|\delta(s) \cap K|$ is even. Then, $s \in T$. This implies that $|(S - \{s\}) \cap T|$ is even. Similarly, $\sum_{v \in S} |\delta(v) \cap K|$ is even. \square

3.2 LP-based $\frac{3}{2}$ -approximation algorithm

In this section, we give an LP-based $\frac{3}{2}$ -approximation algorithm for the s - t path graph-TSP. Before stating the algorithm, we need some lemmas.

Lemma 3.2.1 *There is a polynomial-time algorithm to find all narrow cuts Q_1, Q_2, \dots, Q_k .*

Proof. Compute the Gomory-Hu tree for the terminal vertex set V with respect to the capacity x^* (see [22, Section 3.5.2]). After that, for each edge of the s - t path in the Gomory-Hu tree,

check the corresponding cut. We claim that each such cut with x^* capacity less than 2 is a narrow cut, and there are no other narrow cuts. The correctness of this claim follows from the following observation: For any $u \in L_i, v \in L_{i+1}$, the narrow cut Q_i is the unique minimum u - v cut. So, there exists an edge e_i corresponding to Q_i in the u - v path in the Gomory-Hu tree. And furthermore, Q_i is also an s - t cut. This implies e_i must be in the s - t path in the Gomory-Hu tree. Therefore, to find narrow cuts, we only need to check the cuts corresponding to the edges in the s - t path in the Gomory-Hu tree. \square

Let H^{sup} be the support graph of x^* . For any $L \subseteq V(H^{sup})$, the subgraph of H^{sup} induced by L is denoted by $H^{sup}(L)$. The following is a key lemma in this section.

Lemma 3.2.2 *For $1 \leq p \leq q \leq k+1$, $H^{sup}(\cup_{p \leq i \leq q} L_i)$ is connected.*

Proof. Consider the graph H^{sup} which is the support graph of x^* . Note that $x^*(\delta_{H^{sup}}(S)) = x^*(\delta_H(S))$ for any $\phi \subsetneq S \subsetneq V$. In this proof, the notation refers to H^{sup} , e.g., $\delta(S)$ means $\delta_{H^{sup}}(S)$. Fix p and q , and let $L = \cup_{p \leq i \leq q} L_i$. We divide the proof into several cases:

Case 1: $p = 1$ and $q = k+1$, i.e., $H^{sup} = H^{sup}(L)$. Since the first constraint in **LP1** implies that $x^*(\delta(S)) \geq 1$ for each $\phi \subsetneq S \subsetneq V$, we see that H^{sup} is connected.

Case 2: $p = 1$ and $q < k+1$. Suppose $H^{sup}(L)$ is not connected. Then, there exist two nonempty vertex sets U_1 and U_2 such that U_1, U_2 is a partition of L and there exists no edge between U_1 and U_2 in H^{sup} . Without loss of generality, we can assume that $s \in U_1$. By the constraints of **LP1**, we have $x^*(\delta(U_1)) \geq 1$ and $x^*(\delta(U_2)) \geq 2$. However, $L = Q_q$ is a narrow cut, which implies $x^*(\delta(L)) < 2$. Note that $\delta(U_1) \cap \delta(U_2) = \phi$ and $\delta(L) = \delta(U_1) \cup \delta(U_2)$. So, $2 > x^*(\delta(L)) = x^*(\delta(U_1)) + x^*(\delta(U_2)) \geq 1 + 2 = 3$. This is a contradiction.

Case 3: $p > 1$ and $q = k+1$. By the symmetry of s and t , it is the same as Case 2.

Case 4: $p > 1$ and $q < k+1$. Suppose $H^{sup}(L)$ is not connected. Then, similarly, there exist two nonempty vertex sets U_1 and U_2 such that U_1, U_2 is a partition of L and there exists no edge between U_1 and U_2 in H^{sup} . In this case, by the constraints of **LP1**, we have $x^*(\delta(U_1)) \geq 2$ and $x^*(\delta(U_2)) \geq 2$. Let $Y_1 = \cup_{1 \leq i < p} L_i$ and $Y_2 = \cup_{q < i \leq k+1} L_i$. Note that Y_1 and Y_2 are two narrow cuts. Also, $\delta(U_1) \cup \delta(U_2) \subseteq \delta(Y_1) \cup \delta(Y_2)$. Note that $\delta(U_1) \cap \delta(U_2) = \phi$ by the definition of U_1 and U_2 . Thus, $4 > x^*(\delta(Y_1)) + x^*(\delta(Y_2)) \geq x^*(\delta(Y_1) \cup \delta(Y_2)) \geq x^*(\delta(U_1) \cup \delta(U_2)) = x^*(\delta(U_1)) + x^*(\delta(U_2)) \geq 2 + 2 = 4$. This is a contradiction. \square

By setting $p = q = i$, and by setting $q = p + 1 = i + 1$, we obtain the following corollary:

Corollary 3.2.3 *For each i such that $i = 1, \dots, k+1$, the graph $H^{sup}(L_i)$ is connected, and moreover, there exists an edge connecting L_i and L_{i+1} in H^{sup} .*

In our LP-based approximation algorithm for the s - t path graph-TSP, Lemma 3.2.1 provides a polynomial-time algorithm for Step 2, and Corollary 3.2.3 guarantees that Step 3 and Step 4 are feasible. Thus, the LP-based algorithm runs in polynomial time.

Algorithm 3.1: LP-based approximation algorithm for the s - t path graph-TSP

- 1 Find an optimal solution x^* of **LP1** and construct the support graph H^{sup} of x^* ;
- 2 Find the narrow cuts Q_1, Q_2, \dots, Q_k containing s , and get the corresponding sets L_1, L_2, \dots, L_{k+1} (recall: $L_i = Q_i - Q_{i-1}$ where $Q_0 = \phi$ and $Q_{k+1} = V$). If no narrow cuts exist, take J_{good} as a spanning tree in H and go to Step 6;
- 3 For $1 \leq i \leq k+1$, find a spanning tree J_i on $H^{sup}(L_i)$;
- 4 Take an edge e_i from H^{sup} connecting L_i to L_{i+1} for $1 \leq i \leq k$. Let $E_b = \cup_{1 \leq i \leq k} \{e_i\}$;
- 5 Construct the spanning tree $J_{good} = (\cup_{1 \leq i \leq k+1} J_i) \cup E_b$;
- 6 Let T be the wrong degree vertex set of J_{good} . Find a minimum-size T -join F_{good} in H ;
- 7 Output $J_{good} \dot{\cup} F_{good}$ (disjoint union of edge sets in $2H$);

Lemma 3.2.4 *For the T -join F_{good} in the LP-based approximation algorithm for the s - t path graph-TSP, we have*

$$|F_{good}| \leq \frac{1}{2} \sum_{e \in E(H)} x_e^*.$$

Proof. We claim $x^*(\delta(S)) \geq 2$ for every T -odd cut where T is the wrong degree vertex set of J_{good} in the algorithm. Let S be a T -odd cut. There are two cases to be considered.

Case 1: S is not an s - t cut. Then, by the constraint of **LP1**, we have $x^*(\delta(S)) \geq 2$.

Case 2: S is an s - t cut. If there does not exist any narrow cuts, then clearly $x^*(\delta(S)) \geq 2$. Otherwise, for any narrow cut Q , we have $|J_{good} \cap \delta(Q)| = 1$ by Step 4 of the algorithm. However, by Lemma 3.1.2, we have $|J_{good} \cap \delta(S)|$ is even. This means S is not a narrow cut. Thus, $x^*(\delta(S)) \geq 2$.

By the claim, we know $\frac{1}{2}x^*(\delta(S)) \geq 1$ for every T -odd cut S . This implies $\frac{1}{2}x^*$ is a feasible solution of **LP2**. By Lemma 2.1.1, we have $|F_{good}| \leq \frac{1}{2} \sum_{e \in E(H)} x_e^*$. This completes the proof. \square

Remark 3.2.5 *In fact, if we can find a spanning tree J such that $|J \cap \delta(Q)|$ is odd for each narrow cut Q , then we can find an edge set F to correct the wrong degree vertices in J such that $|F| \leq \frac{1}{2} \sum_{e \in E(H)} x_e^*$.*

Theorem 3.2.6 *The LP-based approximation algorithm for the s - t path graph-TSP achieves an approximation factor of $\frac{3}{2}$.*

Proof. Note that J_{good} is a spanning tree of H . We consider J_{good} as an edge set. So, $|J_{good}| = |V| - 1 \leq \sum_{e \in E(H)} x_e^* \leq PTSP_{opt}(H)$. Also note that $|F_{good}| \leq \frac{1}{2} \sum_{e \in E(H)} x_e^* \leq$

$\frac{1}{2}PTSP_{opt}(H)$ by Lemma 3.2.4. Since $J_{good} \dot{\cup} F_{good}$ is a connected spanning subgraph of $2H$ with $\{s, t\}$ as the odd-degree vertex set, this gives a Hamiltonian s - t path on the metric completion of H with cost at most $|J_{good}| + |F_{good}|$. Therefore, the LP-based algorithm is a $\frac{3}{2}$ -approximation algorithm. \square

Remark 3.2.7 By the proof of Theorem 3.2.6, we can obtain an upper bound of $\frac{3}{2}$ for the integrality ratio of LP1. Furthermore, this also implies that the integrality ratio of the path Held-Karp LP relaxation is at most $\frac{3}{2}$ when restricted to the graphic metric (see Section 4.3). Note that [1, Figure 1(b)] presented an example with graphic metric giving a lower bound of $\frac{3}{2}$ for the integrality ratio of the path Held-Karp LP relaxation. Hence, from this point of view, our algorithm achieves the best possible approximation guarantee that an algorithm can get based on the path Held-Karp LP relaxation for the s - t path graph-TSP.

Chapter 4

Path Traveling Salesman Problem

Given a complete graph with nonnegative metric edge costs and two fixed vertices s, t , the goal of the (metric) s - t path TSP is to find a minimum-cost Hamiltonian path from s to t . Hoogeveen [39] gave an s - t path TSP variant of Christofides' approximation algorithm for TSP [20], and obtained an approximation factor of $\frac{5}{3}$. There was no improvement in this approximation factor for over two decades until An, Kleinberg, and Shmoys [1] improved the approximation factor to $\frac{1+\sqrt{5}}{2} \approx 1.61803$. One of the key new contributions of [1] is to design and analyse a randomized version of Christofides' algorithm. The analysis introduced the notion of a correction vector for the s - t path TSP. Later, Sebő [61] further improved the analysis and obtained a better approximation factor of $\frac{8}{5}$. We mention that Vygen [67], very recently, obtained an approximation factor slightly better than $\frac{8}{5}$. [61] introduced a correction vector different from that of [1], and this is one reason why the analysis in [61] gives a better approximation factor. Informally speaking, a better correction vector provides a better approximation factor.

In this chapter, ¹ we give a unified presentation of the results from both [1] and [61] by introducing a new correction vector that we call the *unified correction vector*. Our correction vector is simple and it leads to short derivations of the approximation factors of both [1] and [61]. The difference between our correction vector and the previous ones is that it assigns the value one to the minimum-cost edge in each so-called τ -narrow cut, whereas the correction vectors used in [1] and [61] are fractional on each τ -narrow cut. We mention that Vygen's [66] comprehensive recent survey discusses the common points of the analysis of [1] and [61], and the survey sketches short proofs of both approximation factors; however, [66] uses the same correction vectors as [1] and [61].

¹ The contents of this chapter appear in the submitted preprint [34]. This preprint has been accepted for publication in SIAM Journal on Discrete Mathematics.

An et al. [1] and Sebő [61] use two different LP relaxations of the s - t path TSP in their algorithms. [1] uses the path version of the Held-Karp LP relaxation for TSP, whereas [61] uses a weaker LP relaxation, the Partition LP in Section 2.1.1. This motivates a comparison of these two LP relaxations. We mention that Sebő proves an approximation factor of $\frac{8}{5}$ for a more general problem, namely the *connected T -join problem*, and the LP in his paper is a relaxation of this problem. We show that both LPs for the s - t path TSP have the same (fractional) optimal value. Also, we show that the minimum cost of integral solutions of the two LPs are within a factor of $\frac{3}{2}$ of each other; moreover, we present an example to show that the factor of $\frac{3}{2}$ is tight. We prove this result by showing that a half-integral solution of the stronger LP-relaxation of cost c can be rounded to an integral solution of cost at most $\frac{3}{2}c$.

For the s - t path TSP, it is known that the integrality ratio of the path Held-Karp LP relaxation has a lower bound of $\frac{3}{2}$. The algorithms from [1] and [61] mentioned above are LP-based. This leads to the upper bound $\frac{8}{5}$ on the integrality ratio of the LP relaxation. A natural open question is to close this gap by designing an LP-based $\frac{3}{2}$ -approximation algorithm for the s - t path TSP. For the s - t path graph-TSP, a critical special case of the s - t path TSP, the integrality ratio of the corresponding LP relaxation has been resolved already. The first $\frac{3}{2}$ -approximation algorithm was given by Sebő and Vygen [62] using ear decompositions. In Chapter 3, we presented another, conceptually simpler, LP-based $\frac{3}{2}$ -approximation algorithm. The analysis of the $\frac{3}{2}$ -approximation factor of Chapter 3 uses the graphic property only for one point: to guarantee that the cost of a special spanning tree constructed in the algorithm is at most the optimum of the LP relaxation. A natural question is whether we can extend this graphic LP-based approximation algorithm and analysis to the general metric case. Unfortunately, we present an instance that shows that the natural extension is not possible. Moreover, our instance also illustrates that probabilistic methods are relevant for the analysis of improved LP-based approximation algorithms. This instance may shed some light on how to design a better approximation algorithm for the s - t path TSP.

This chapter is organized as follows. Section 4.1 has some notation and basic results. Section 4.2 presents our unified correction vector. Section 4.3 shows the relationship of two different LP relaxations of the s - t path TSP. Section 4.4 discusses an instance that points to some of the obstructions for obtaining better approximation factors.

4.1 Preliminaries

Let $G = (V, E)$ be a complete graph with metric edge costs c . Let s, t be two fixed vertices in G . When there is no risk of confusion, we will use the same notation H for a subgraph H and its edge set $E(H)$.

For any probabilistic event A , we use $\Pr(A)$ to denote the probability of occurrence of A . For a random variable R , the expectation of R is denoted by $\mathbb{E}(R)$.

4.1.1 Linear programs

Recall from Section 2.1.1 that the path Held-Karp LP relaxation for the s - t path TSP is defined as follows:

$$\begin{aligned}
& \text{minimize : } \sum_{e \in E} c_e x_e & (\text{LP3}) \\
& \text{subject to : } x(\delta(s)) = x(\delta(t)) = 1 \\
& \quad x(\delta(v)) = 2 & \forall v \neq s, t \\
& \quad x(\delta(S)) \geq 1 & \forall s\text{-}t \text{ cut } S \\
& \quad x(\delta(S)) \geq 2 & \forall \emptyset \subsetneq S \subsetneq V, |S \cap \{s, t\}| \text{ even} \\
& \quad 1 \geq x_e \geq 0 & \forall e \in E
\end{aligned}$$

The spanning tree polytope is shown as follows:

$$\begin{aligned}
& \text{minimize : } \sum_{e \in E} c_e x_e & (\text{LP4}) \\
& \text{subject to : } x(E) = |V| - 1 \\
& \quad x(E(S)) \leq |S| - 1 & \forall \emptyset \subsetneq S \subsetneq V \\
& \quad x_e \geq 0 & \forall e \in E
\end{aligned}$$

Lemma 4.1.1 *Every solution x of LP3 lies in the spanning tree polytope LP4.*

Proof. By the degree constraint for each vertex in LP3, we have $x(E) = |V| - 1$. Now consider the second set of constraints in LP4. If $|S \cap \{s, t\}|$ is even, by the degree and cut constraints in LP3, $x(E(S)) = \frac{\sum_{v \in S} x(\delta(v)) - x(\delta(S))}{2} \leq \frac{2|S| - 2}{2} = |S| - 1$. Otherwise, $|S \cap \{s, t\}| = 1$, in which case, $x(E(S)) = \frac{\sum_{v \in S} x(\delta(v)) - x(\delta(S))}{2} \leq \frac{(2|S| - 1) - 1}{2} = |S| - 1$. This completes the proof. \square

4.1.2 T -joins

Recall from Section 2.1.1 that the following LP is the T -join polyhedron that formulates the problem of finding a T -join of minimum cost on G :

$$\begin{aligned}
 & \text{minimize : } \sum_{e \in E} c_e x_e & (\text{LP5}) \\
 & \text{subject to : } x(\delta(S)) \geq 1 & \forall T\text{-odd } S \\
 & & x_e \geq 0 \quad \forall e \in E
 \end{aligned}$$

Recall from Section 3.1 that for any spanning tree K , the set of wrong degree vertices of K is defined as $\{v \in \{s, t\} : |\delta(v) \cap K| \text{ even}\} \cup \{v \in V - \{s, t\} : |\delta(v) \cap K| \text{ odd}\}$.

4.1.3 Polyhedra and convex decomposition

Let

$$P := \{x : Ax \leq b\} \quad \text{where } A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m.$$

Let x' be a feasible solution of P . For a constraint $a_i^\top x \leq b_i$ in P , we say x' is *tight* at this constraint if $a_i^\top x' = b_i$. Let x_1, x_2 be two distinct feasible solutions of P . If there exists a $0 < \lambda < 1$ and $y \in P$ such that $\lambda x_1 + (1 - \lambda)y = x_2$, we say x_1 is *in some convex decomposition* of x_2 in P .

From the geometry of polyhedra, we have the following characterization of the convex decompositions.

Lemma 4.1.2 *The solution x_1 is in some convex decomposition of x_2 in P if and only if x_1 is tight at the constraints of P where x_2 is tight.*

A nonempty set $F \subseteq P$ is a face if and only if there exists an index set $I \subseteq \{1, 2, \dots, m\}$ such that $F = \{x : a_i^\top x = b_i \text{ for } i \in I, a_i^\top x \leq b_i \text{ for } i \notin I\}$ where $a_i^\top x \leq b_i$ is the i th constraint of P (see [22, Section 6.2]). Hence, the solution x_1 is in some convex decomposition of x_2 in P if and only if x_1 is in the minimal face of P that contains x_2 .

4.1.4 Christofides' algorithm for s - t path TSP

Hoogeveen [39] gave a variant of Christofides' algorithm to achieve the first approximation factor of $\frac{5}{3}$ for the s - t path TSP.

Compute a minimum-cost spanning tree J^* . Let T be the set of wrong degree vertices of J^* . Find a minimum-cost T -join F^* . Then, the union $J^* \dot{\cup} F^*$ of J^* and F^* (that keeps the duplicated edges) forms a connected graph that has even degree at all nodes except s and t . One can then take the Eulerian traversal that starts at s and ends at t , and shortcut it, to obtain an s - t path visiting all vertices of no greater cost.

Theorem 4.1.3 [39] *Christofides' algorithm for s - t path TSP achieves an approximation factor of $\frac{5}{3}$.*

We present a nice proof from Sebő and Vygen [62].

Proof. Let P^* be an optimal solution of s - t path TSP. Let T, J^*, F^* be as in the algorithm. Let R be the s - t path in J^* . Since P^* is a Hamiltonian path from s to t , we can extract from it a subset of edges, F_{P^*} , that forms a T -join by pairing successive vertices of T in the path. Since P^* is a spanning tree, we know $c(J^*) \leq c(P^*)$. Note that we only need to prove $c(F^*) \leq \frac{2}{3}c(P^*)$. This follows from the fact that $J^* \dot{\cup} P^*$ can be partitioned into three T -joins: one is $J^* - R$, one is F_{P^*} , and one is the union of R and $P^* - F_{P^*}$. One can check that each of these edge sets is a T -join by using the fact that T is the set of wrong degree vertices of J^* . Then, $3c(F^*) \leq c(J^*) + c(P^*) \leq 2c(P^*)$. This completes the proof. \square

4.2 Unified correction vector

An et al. [1] designed a randomized Christofides' algorithm for the s - t path TSP, and they proved an approximation factor of $\frac{1+\sqrt{5}}{2}$ by analysing this algorithm. Their algorithm and their analysis were based on the LP relaxation LP3. Sebő [61] presented a new analysis of this randomized algorithm and improved the approximation factor to $\frac{8}{5}$. The algorithm and analysis of [61] were based on a different LP relaxation, see the Partition LP in Section 2.1.1. The Partition LP is restated as LP6 in Section 4.3. In Section 4.3, we prove that LP3 and LP6 have the same optimal value. This result together with a few more observations implies that LP6 can be replaced by LP3 in the algorithm and analysis of [61] to achieve the same approximation factor of $\frac{8}{5}$. In this

section, we prove the approximation factor of [1]; also, we prove the $\frac{8}{5}$ -approximation factor of [61] based on LP3 rather than LP6.

Randomized Christofides' algorithm:

Solve the LP relaxation LP3 to get an optimal solution x^* . Since x^* is in the spanning tree polytope, there exists a convex decomposition of spanning trees J_1, \dots, J_l such that $\sum_{1 \leq i \leq l} \lambda_i \mathcal{X}^{J_i} = x^*$ where $\sum_{1 \leq i \leq l} \lambda_i = 1$, $\lambda_i > 0$ and \mathcal{X}^{J_i} is the edge incidence vector of J_i . Such a decomposition can be found in polynomial time, see Theorem 51.5 of [60]. We sample a spanning tree J from these spanning trees according to the probability defined by the coefficient λ_i of each spanning tree in the convex combination. Let T denote the set of the wrong degree vertices of J . Then, as in Christofides' algorithm, a minimum-cost T -join F is added to fix the wrong degree vertices of J .

The expected cost of the random solution of the algorithm is the sum of the expected cost of J , which is the cost of x^* , and the expected cost of the T -join F . Any feasible solution of the T -join polyhedron provides a cost upper bound for the T -join F . We call a feasible solution to the T -join polyhedron a *fractional T -join*. An et al. [1] introduced correction vectors to construct a special type of fractional T -join. A *correction vector* for a τ -narrow cut S is an edge vector z that satisfies $\sum_{e \in \delta(S)} z_e \geq 1$, where the definition of τ -narrow cut will be given next. The correction vectors were further analyzed in [61] to obtain a better approximation factor. In this section, we present a unified correction vector to derive the results of both [1] and [61].

The following key definition is introduced in [1]. Let $0 < \tau \leq 1$. If an s - t cut Q satisfies $x^*(\delta(Q)) < 1 + \tau$, we call it a τ -narrow cut (for the solution x^* of LP3). Let \mathcal{C}_τ be the set of all τ -narrow cuts that contain s . The τ -narrow cuts have the same nice property as the narrow cuts in Section 3.1 although they are defined for the feasible solutions of different LP relaxations. In fact, a narrow cut can be considered as 1-narrow cut. The proof of the following lemma is the same as the proof of Lemma 3.1.1.

Lemma 4.2.1 [1] *Let Q_1, Q_2 be two distinct cuts in \mathcal{C}_τ (for the solution x^* of LP3). Then either $Q_1 \subsetneq Q_2$ or $Q_2 \subsetneq Q_1$.*

Similarly to the narrow cuts in Section 3.1, we can use Q_1, Q_2, \dots, Q_k to denote all of the τ -narrow cuts containing s such that $s \in Q_1 \subsetneq Q_2 \subsetneq Q_3 \cdots \subsetneq Q_k \subsetneq V$. Note that $\mathcal{C}_\tau = \{Q_i\}_{1 \leq i \leq k}$. Define $L_i = Q_i - Q_{i-1}$ for $i = 1, 2, \dots, k, k+1$ where $Q_0 = \emptyset$ and $Q_{k+1} = V$. Each L_i is nonempty and $\cup_{1 \leq i \leq k+1} L_i = V$. We call $\{L_i\}$ the *partition derived by the τ -narrow cuts \mathcal{C}_τ* .

Let \mathcal{X}^J denote the edge incidence vector of the edge set of J . For any $Q \in \mathcal{C}_\tau$, we let e_Q be an edge in $\delta(Q)$ of minimum cost. Let \mathcal{X}^{e_Q} denote the edge incidence vector of $\{e_Q\}$, i.e.,

$\mathcal{X}_{e_Q}^{e_Q} = 1$, and $\mathcal{X}_e^{e_Q} = 0$ if $e \neq e_Q$. Our *unified correction vector* is defined as \mathcal{X}^{e_Q} for each $Q \in \mathcal{C}_\tau$, i.e., the unified correction vector simply assigns the value one to the minimum-cost edge in each τ -narrow cut. In contrast, the correction vectors used in [1] and [61] are fractional but sum up to at least one for each τ -narrow cut.

Let α, β and τ be real parameters between 0 and 1, whose specific values are given later. Recall that J is the random spanning tree in the randomized Christofides' algorithm. Our fractional feasible T -join solution with unified correction vectors, called *unified fractional T -join*, is as follows:

Unified fractional T -join:

$$f = \alpha \mathcal{X}^J + \beta x^* + \sum_{Q \in \mathcal{C}_\tau, Q \text{ is } T\text{-odd}} (1 - 2\alpha - \beta x^*(\delta(Q))) \mathcal{X}^{e_Q},$$

where α, β, τ satisfy the following condition:

$$\alpha + 2\beta = 1, \tau = \frac{1 - 2\alpha}{\beta} - 1, \alpha \geq 0 \text{ and } \beta \geq 0. \quad (4.1)$$

Let us derive the settings of α, β and τ in (4.1). The purpose of the unified fractional T -join f is to provide an upper bound on the cost of the minimum-cost T -join F in the randomized Christofides' algorithm. By Lemma 2.1.1, it suffices to make f feasible for the T -join polyhedron LP5. This requires special settings of α, β and τ .

Consider the cut constraints in LP5. Let S be a T -odd cut. First we need to make sure that for any $Q \in \mathcal{C}_\tau$, the coefficient $1 - 2\alpha - \beta x^*(\delta(Q))$ is nonnegative. Since $x^*(\delta(Q)) < 1 + \tau$ for any $Q \in \mathcal{C}_\tau$, it suffices to set $1 - 2\alpha - \beta(1 + \tau) = 0$, i.e., $\tau = \frac{1 - 2\alpha}{\beta} - 1$.

Suppose that S is an s - t cut. Note that S is T -odd. Hence, by Lemma 3.1.2, $|\delta(S) \cap J|$ is even. If S is not a τ -narrow cut, then $f(\delta(S)) \geq \alpha \mathcal{X}^J(\delta(S)) + \beta x^*(\delta(S)) \geq 2\alpha + \beta(1 + \tau)$. By the assumption that $\tau = \frac{1 - 2\alpha}{\beta} - 1$, we have $f(\delta(S)) \geq 1$ in this case. If S is a cut in \mathcal{C}_τ , then $f(\delta(S)) \geq 2\alpha + \beta x^*(\delta(S)) + (1 - 2\alpha - \beta x^*(\delta(S))) \mathcal{X}^{e_S}(\delta(S)) \geq 1$.

Now the only remaining case is that S is $\{s, t\}$ -even. Then $x^*(\delta(S)) \geq 2$ by LP3. Since J is a spanning tree, we have $\mathcal{X}^J(\delta(S)) \geq 1$. This implies $f(\delta(S)) \geq \alpha \mathcal{X}^J(\delta(S)) + \beta x^*(\delta(S)) \geq \alpha + 2\beta$. Hence, in this case, it suffices to set $\alpha + 2\beta = 1$.

Hence, we have the following result by the analysis above.

Lemma 4.2.2 *The unified fractional T -join f is a feasible solution of the T -join polyhedron LP5.*

Lemma 4.2.2 shows that the expected cost of the minimum-cost T -join F computed by the randomized Christofides' algorithm is at most the expected cost of the unified fractional T -join. Hence, the expected cost of the solution of the randomized Christofides' algorithm is upper bounded by the optimal value of LP3 plus the expected cost of the unified fractional T -join. In Section 4.2.1 and Section 4.2.2, we will present two different analyses of the expected cost of the unified fractional T -join to derive two different approximation factors from [1] and [61] for the randomized Christofides' algorithm.

Remark 4.2.3 *From the analysis above, the cost analysis of the unified fractional T -join is critical for proving an approximation factor for the randomized Christofides' algorithm. If we can get a better upper bound on the cost of the unified fractional T -join, then the approximation factor can be further improved.*

The following lemma is used in the analysis of the expected cost of the unified fractional T -join in Section 4.2.1 and Section 4.2.2.

Lemma 4.2.4 [1][61] *Let J be the random spanning tree and T be the set of wrong degree vertices of J in the randomized Christofides' algorithm. Let $Q \in \mathcal{C}_\tau$, i.e., Q is a τ -narrow cut. Then*

- (i) $\Pr(|\delta(Q) \cap J| = 1) \geq 2 - x^*(\delta(Q))$, and
- (ii) $\Pr(Q \text{ is } T\text{-odd}) \leq x^*(\delta(Q)) - 1$.

Proof. Since J is a spanning tree, $|\delta(Q) \cap J| \geq 1$ always holds. So $\sum_{i \geq 1} \Pr(|\delta(Q) \cap J| = i) = 1$. Then

$$\begin{aligned}
 \Pr(|\delta(Q) \cap J| \geq 2) &\leq \sum_{i \geq 1} i * \Pr(|\delta(Q) \cap J| = i) - \sum_{i \geq 1} \Pr(|\delta(Q) \cap J| = i) \\
 &= \mathbb{E}(|\delta(Q) \cap J|) - \sum_{i \geq 1} \Pr(|\delta(Q) \cap J| = i) \\
 &= x^*(\delta(Q)) - 1.
 \end{aligned}$$

Note that $\mathbb{E}(|\delta(Q) \cap J|) = x^*(\delta(Q))$ follows from the fact that $\mathbb{E}(\mathcal{X}^J) = x^*$ since J is a random tree in the convex decomposition of spanning trees for x^* where the coefficients of the spanning trees define the probability distribution. Thus, we have $\Pr(|\delta(Q) \cap J| = 1) = 1 - \Pr(|\delta(Q) \cap J| \geq 2) \geq 2 - x^*(\delta(Q))$. This proves the first inequality.

Now consider the second inequality. By Lemma 3.1.2, $|\delta(Q) \cap J|$ is even if Q is T -odd. This means $\Pr(Q \text{ is } T\text{-odd}) \leq \Pr(|\delta(Q) \cap J| \text{ is even}) \leq \Pr(|\delta(Q) \cap J| \geq 2) \leq x^*(\delta(Q)) - 1$. \square

4.2.1 AKS' $\frac{1+\sqrt{5}}{2}$ -approximation via unified correction vector

First, we present two lemmas needed for the cost analysis of the randomized Christofides' algorithm.

Lemma 4.2.5 *Let K be a spanning tree with n vertices and edge set $E(K)$. Let $\mathcal{S} = \{S_i : 1 \leq i \leq n-1\}$ be a family of subsets of the vertex set of K such that $|S_i| = i$ and $S_i \subsetneq S_{i+1}$. There exists a bijection from \mathcal{S} to $E(K)$ such that each cut S_i is mapped to an edge of K in $\delta(S_i)$.*

Proof. Without loss of generality, we can assume that the vertex set of K is $\{v_1, v_2, \dots, v_n\}$ and $S_i = \{v_1, v_2, \dots, v_i\}$ for $1 \leq i \leq n-1$. We prove the result by induction on n . The statement is clearly true for $n = 2$. Suppose $n \geq 3$. Consider the vertex v_n .

We first pick the edge e of K incident with v_n in the unique path of K between v_{n-1} and v_n . We map S_{n-1} to this edge e . Let K' be the graph obtained from $K - \{e\}$ by contracting v_{n-1} and v_n into a single vertex v'_{n-1} . Note that K' is a connected graph with $n-2$ edges. This implies that K' is a spanning tree with $n-1$ vertices $\{w_1, w_2, \dots, w_{n-1}\}$ where $w_i = v_i$ for $1 \leq i \leq n-2$ and $w_{n-1} = v'_{n-1}$. Note that $\delta(\{w_1, w_2, \dots, w_i\})$ is a subset of $\delta(S_i)$ for $1 \leq i \leq n-2$. Hence, we can define the rest of the bijection by applying the induction hypothesis to the spanning tree K' on these $n-1$ vertices.

□

Lemma 4.2.6 *We have*

$$\sum_{Q \in \mathcal{C}_\tau} c(e_Q) \leq c(x^*). \quad (4.2)$$

Proof. Let K_{min} be a minimum-cost spanning tree on G . Consider the partition $\{L_i\}$ derived by \mathcal{C}_τ . We contract every L_i into a single vertex. Then the resulting graph obtained from K_{min} is connected. Let K be a spanning tree of the contracted graph. Applying Lemma 4.2.5 to K , we construct an injective mapping ϕ from \mathcal{C}_τ to the edge set of K such that $\phi(Q) \in \delta(Q)$ for each $Q \in \mathcal{C}_\tau$. Note that $K \subseteq K_{min}$. Then $\sum_{Q \in \mathcal{C}_\tau} c(e_Q) \leq \sum_{Q \in \mathcal{C}_\tau} c(\phi(Q)) \leq c(K_{min}) \leq c(x^*)$ since x^* is in the spanning tree polytope. The first inequality follows from the fact that e_Q is the minimum-cost edge in $\delta(Q)$. □

Theorem 4.2.7 [1] *The randomized Christofides' algorithm achieves an approximation factor of $\frac{1+\sqrt{5}}{2}$.*

Proof. Since J is a random spanning tree based on the convex decomposition of spanning trees for x^* , we have $\mathbb{E}(\mathcal{X}^J) = x^*$. Hence, the expected cost of the solution of the randomized Christofides' algorithm is upper bounded by the optimal value of **LP3** plus the expected cost of the minimum-cost T -join F . By Lemma 2.1.1 and Lemma 4.2.2, the expected cost of F is at most the expected cost of the unified fractional T -join.

$$\begin{aligned}
& \mathbb{E}[c(\alpha \mathcal{X}^J + \beta x^* + \sum_{Q \in \mathcal{C}_\tau, Q \text{ is } T\text{-odd}} (1 - 2\alpha - \beta x^*(\delta(Q))) \mathcal{X}^{e_Q})] \\
& \stackrel{\text{Lemma 4.2.4}}{\leq} (\alpha + \beta)c(x^*) + \sum_{Q \in \mathcal{C}_\tau} (x^*(\delta(Q)) - 1)(1 - 2\alpha - \beta x^*(\delta(Q)))c(e_Q) \\
& \leq (\alpha + \beta)c(x^*) + \max_{0 \leq z < \tau} z(1 - 2\alpha - \beta z - \beta) \sum_{Q \in \mathcal{C}_\tau} c(e_Q) \\
& \stackrel{\text{Lemma 4.2.6}}{\leq} (\alpha + \beta + \max_{0 \leq z < \tau} z(1 - 2\alpha - \beta z - \beta))c(x^*) \\
& \stackrel{\text{By (4.1)}}{=} (\alpha + \beta + \beta \max_{0 \leq z < \tau} z(\tau - z))c(x^*).
\end{aligned}$$

The last equality follows from the fact that $1 - 2\alpha = \beta(\tau + 1)$ by (4.1). The value of z that maximizes the expression is $\frac{\tau}{2}$. Hence, the upper bound on the expected cost of the unified fractional T -join is at most $(\alpha + \beta + \beta(\frac{\tau}{2})^2)c(x^*)$. Substitute $\tau = \frac{1-2\alpha}{\beta} - 1$, $\alpha = 1 - 2\beta$ from (4.1) into the upper bound. Minimizing with respect to β gives $\frac{\sqrt{5}-1}{2}c(x^*)$ with optimal settings: $\beta = \frac{1}{\sqrt{5}}$, $\alpha = 1 - \frac{2}{\sqrt{5}}$, $\tau = 3 - \sqrt{5}$. Therefore, the optimal value of **LP3** plus this upper bound $\frac{\sqrt{5}-1}{2}c(x^*)$ on the expected cost of the unified fractional T -join leads to the approximation factor of $\frac{1+\sqrt{5}}{2}$ that was first proved in [1]. \square

In [1], the correction vector is constructed by using flow computations to map the optimal LP solution x^* to the τ -narrow cuts. In contrast, our unified correction vector simply assigns the value one to the minimum-cost edge in each τ -narrow cut. We avoid the flow computation argument of [1] by using Lemma 4.2.5.

4.2.2 Sebő's $\frac{8}{5}$ -approximation via unified correction vector

Let P be the s - t path in J . Sebő [61] points out the crucial fact that $J - P$ is a T -join for the set of wrong degree vertices T of J . Recall that F is the minimum-cost T -join in the randomized Christofides' algorithm. This implies that $\mathbb{E}(c(F)) \leq \mathbb{E}(c(J - P))$. Note that $c(x^*) = \mathbb{E}(c(J)) = \mathbb{E}(c(J - P)) + \mathbb{E}(c(P))$.

It turns out that $\mathbb{E}(c(P))$ also serves as an upper bound in another cost inequality similar to (4.2); see the following lemma.

Lemma 4.2.8 *We have*

$$\sum_{Q \in \mathcal{C}_\tau} (2 - x^*(\delta(Q)))c(e_Q) \leq \mathbb{E}(c(P)). \quad (4.3)$$

Proof. Let $Q \in \mathcal{C}_\tau$; thus, Q is a τ -narrow cut. If $|\delta(Q) \cap J| = 1$, then let e'_Q denote the unique edge in $\delta(Q) \cap J$. Recall that a τ -narrow cut is an s - t cut, and therefore e'_Q must be in P since P is the s - t path in J . Moreover, observe that Q is one of the two connected components of $J - \{e'_Q\}$. Hence, for distinct $Q_1, Q_2 \in \mathcal{C}_\tau$ such that $|\delta(Q_1) \cap J| = 1$ and $|\delta(Q_2) \cap J| = 1$, the edges e'_{Q_1} and e'_{Q_2} must be distinct (otherwise, $J - \{e'_{Q_1}\}$ and $J - \{e'_{Q_2}\}$ would have the same connected components, contradicting the fact that Q_1, Q_2 are distinct sets containing s). Then

$$c(P) \geq \sum_{|\delta(Q) \cap J|=1, Q \in \mathcal{C}_\tau} c(e'_Q) \geq \sum_{|\delta(Q) \cap J|=1, Q \in \mathcal{C}_\tau} c(e_Q).$$

By Lemma 4.2.4,

$$\mathbb{E}(c(P)) \geq \sum_{Q \in \mathcal{C}_\tau} \Pr(|\delta(Q) \cap J| = 1) c(e_Q) \geq \sum_{Q \in \mathcal{C}_\tau} (2 - x^*(\delta(Q)))c(e_Q).$$

□

Theorem 4.2.9 [61] *The randomized Christofides' algorithm achieves an approximation factor of $\frac{8}{5}$.*

Proof. By an argument similar to the one in the proof of Theorem 4.2.7, we are only concerned with the expected cost of the unified fractional T -join, which bounds the expected cost of the minimum-cost T -join F in the randomized Christofides' algorithm.

$$\begin{aligned} & \mathbb{E}[c(\alpha \mathcal{X}^J + \beta x^* + \sum_{Q \in \mathcal{C}_\tau, Q \text{ is } T\text{-odd}} (1 - 2\alpha - \beta x^*(\delta(Q))) \mathcal{X}^{e_Q})] \\ & \stackrel{\text{Lemma 4.2.4}}{\leq} (\alpha + \beta)c(x^*) + \sum_{Q \in \mathcal{C}_\tau} (x^*(\delta(Q)) - 1)(1 - 2\alpha - \beta x^*(\delta(Q)))c(e_Q) \\ & \leq (\alpha + \beta)c(x^*) + \sum_{Q \in \mathcal{C}_\tau} \frac{(x^*(\delta(Q)) - 1)(1 - 2\alpha - \beta x^*(\delta(Q)))}{2 - x^*(\delta(Q))} (2 - x^*(\delta(Q)))c(e_Q) \end{aligned}$$

$$\begin{aligned}
&\leq (\alpha + \beta)c(x^*) + \max_{0 \leq z < \tau} \frac{z(1 - 2\alpha - \beta z - \beta)}{1 - z} \sum_{Q \in \mathcal{C}_\tau} (2 - x^*(\delta(Q)))c(e_Q) \\
&\stackrel{\text{Lemma 4.2.8}}{\leq} (\alpha + \beta)c(x^*) + \max_{0 \leq z < \tau} \frac{z(1 - 2\alpha - \beta z - \beta)}{1 - z} \mathbb{E}(c(P)) \\
&\stackrel{\text{By (4.1)}}{=} (\alpha + \beta)c(x^*) + \beta \max_{0 \leq z < \tau} \frac{z(\tau - z)}{1 - z} \mathbb{E}(c(P)). \tag{4.4}
\end{aligned}$$

The last equality follows from the fact that $1 - 2\alpha = \beta(\tau + 1)$ by (4.1). The value of z that maximizes the expression is $1 - \sqrt{1 - \tau}$. Hence, the upper bound on the expected cost of the unified fractional T -join is at most $(\alpha + \beta)c(x^*) + \beta(1 - \sqrt{1 - \tau})^2 \mathbb{E}(c(P))$. Substitute $\tau = \frac{1 - 2\alpha}{\beta} - 1$, $\alpha = 1 - 2\beta$ from (4.1) into (4.4). Then the coefficients of the terms in (4.4) only depend on β . Denote the coefficient of the last term in (4.4) by $h(\beta)$ where $h(\beta) = (\sqrt{\beta} - \sqrt{1 - 2\beta})^2$. Then the bound can be written as $(1 - \beta)c(x^*) + h(\beta)\mathbb{E}(c(P))$. Note that $c(x^*) = \mathbb{E}(c(J - P)) + \mathbb{E}(c(P))$. Assume $\mathbb{E}(c(P)) = \lambda_0 c(x^*)$. So $0 \leq \lambda_0 \leq 1$ and $\mathbb{E}(c(J - P)) = (1 - \lambda_0)c(x^*)$. Since $\mathbb{E}(c(J - P)) \geq \mathbb{E}(c(F))$, we have

$$\begin{aligned}
\mathbb{E}(c(F)) &\leq \min\{(1 - \lambda_0)c(x^*), (1 - \beta + h(\beta)\lambda_0)c(x^*)\} \\
&\leq \max_{0 \leq \lambda \leq 1} \{\min\{(1 - \lambda)c(x^*), (1 - \beta + h(\beta)\lambda)c(x^*)\}\}. \tag{4.5}
\end{aligned}$$

λ maximizes the expression when $(1 - \lambda)c(x^*) = (1 - \beta + h(\beta)\lambda)c(x^*)$. So $\lambda = \frac{\beta}{h(\beta) + 1}$. Minimizing the upper bound in (4.5) with respect to β gives $\frac{3}{5}c(x^*)$ with optimal settings: $\beta = \frac{4}{9}$, $\alpha = \frac{1}{9}$, $\tau = \frac{3}{4}$; moreover, $\lambda = \frac{2}{5}$. Therefore, the optimal value of LP3 plus this upper bound $\frac{3}{5}c(x^*)$ leads to the approximation factor of $\frac{8}{5}$ that was first proved in [61]. \square

4.3 Linear programming relaxations of the s - t path TSP

In this section, we investigate the relationship between two different LP relaxations of the s - t path TSP. Let $H = (V, E(H))$ be a connected graph with nonnegative edge costs c^H , and let s and t be two fixed vertices. Let $G = (V, E)$ be the metric completion of H with metric costs c . As mentioned in Section 4.1.1, LP3 is a linear programming relaxation of the s - t path TSP on G .

Recall from Section 2.1.1 that the s - t path TSP on G is equivalent to finding a minimum-cost connected spanning subgraph of $2H$ with $\{s, t\}$ as the odd-degree vertex set. The Partition LP defined on H in Section 2.1.1 is an LP relaxation of the s - t path TSP for this equivalent

definition. We restate the Partition LP as follows:

$$\begin{aligned}
& \text{minimize : } \sum_{e \in E(H)} c_e^H x_e & (LP6) \\
& \text{subject to : } x(\delta(\mathcal{W})) \geq |\mathcal{W}| - 1 & \forall \text{ partition } \mathcal{W} \text{ of } V \\
& & x(\delta(S)) \geq 2 & \forall \emptyset \subsetneq S \subsetneq V, |S \cap \{s, t\}| \text{ even} \\
& & x_e \geq 0 & \forall e \in E(H)
\end{aligned}$$

Note that [LP6](#) is defined on the original graph H but [LP3](#) is defined on the metric completion G of H .

In this section, we show that both LPs, [LP3](#) and [LP6](#), have the same (fractional) optimal value, see Corollary 4.3.3. But these two LPs can differ with respect to integral solutions. Observe that the integral solutions of [LP3](#) are exactly the s - t Hamiltonian paths of G ; this follows because an integral solution induces a graph that is connected, has degree one at s, t , and has degree two at all other vertices. The integral solutions of [LP6](#) need not correspond to the s - t Eulerian paths of H ; see the example shown in Figure 4.1.

Let $Opt(LP)$ denote the optimal value of LP, for $LP = \text{LP3}, \text{LP6}$. Let $Opt_{int}(LP)$ denote the minimum cost of an integral solution that satisfies all constraints of LP, for $LP = \text{LP3}, \text{LP6}$. We call $Opt_{int}(LP)$ the *optimal integral value* of a linear program LP. The following table summarizes the relationship between the two LPs; the new results of this section appear in the last two columns.

LPs	Graph	Costs	Optimum	Optimal Integral Value
LP3	G : metric completion of H	c : metric extension of c^H	$Opt(\text{LP3})$	$Opt_{int}(\text{LP3})$
LP6	H	$c^H \geq 0$	$Opt(\text{LP3})$	$Opt_{int}(\text{LP6}) \leq Opt_{int}(\text{LP3}) \leq \frac{3}{2} Opt_{int}(\text{LP6})$

To obtain these results, we need an edge-splitting lemma. Let K be a multigraph, i.e., two adjacent vertices in K may be connected by one or more edges. Let $(u, v), (v, w) \in E(K)$. The *edge-splitting operation* on $(u, v), (v, w)$ at the vertex v is defined as follows:

- Remove $(u, v), (v, w)$ and then add (u, w) if $u \neq w$.

If $u = w$, then we remove the loop formed by adding (u, w) ; note that this removal of the loop has no effect on the edge-connectivity of the graph. We use the following result to prove Lemma 4.3.2; see [31, Theorem A'].

Lemma 4.3.1 [48][49, Ex. 6.51] Let K be a multigraph with even degree at each vertex. Let $v \in V(K)$ and let $U = V(K) - \{v\}$. Let d be a positive integer. If

$$|\delta(S)| \geq d \text{ for each } \emptyset \subsetneq S \subsetneq U \quad (4.6)$$

then the edges incident with v can be partitioned into $\frac{|\delta(v)|}{2}$ disjoint edge pairs (p, v) , (v, q) such that the multigraph obtained by applying the edge-splitting operation to any one of these edge pairs (at the vertex v) still satisfies (4.6).

Lemma 4.3.2 Let x be a rational solution of LP6 of cost $c^H(x)$. Then there exists a solution x' of LP3 with cost at most $c^H(x)$. Moreover, if x is an integral solution, then x' is half-integral.

Proof. The first part of this statement follows from the parsimonious property shown in [8]. However, to show the second part of the statement, we present a proof for the first part as well.

Define an edge vector y on G as follows:

$$y_e = \begin{cases} x_e, & \text{if } e \in E(H), \\ 0, & \text{otherwise.} \end{cases}$$

Since G is the metric completion of H , we know $c(y) \leq c^H(x)$. Then we construct y' from y as follows:

$$y'_e = \begin{cases} 1 + y_e, & \text{if } e = (s, t), \\ y_e, & \text{otherwise.} \end{cases}$$

By the constraints of LP6 and the fact that $y'_{(s,t)} = y_{(s,t)} + 1$, we have $y'(\delta(S)) \geq 2$ for each cut S . Let C be a positive integer such that Cy' is integral. Consider the multigraph K_{2C} with $2Cy'_{(u,v)}$ number of edges between u and v . Then $|\delta_{K_{2C}}(S)| \geq 4C$.

By using Lemma 4.3.1, we apply edge-splitting operations at every vertex until the degree of every vertex is exactly $4C$. We claim that this procedure can be applied such that the number of edges between s and t is $\geq 2C$. To see this, consider an edge-splitting operation at s or t , say s ; note that edge-splitting operation at other vertices does not decrease the number of edges between s and t . There are at least $2C + 1$ feasible edge-splitting pairs available at s (since otherwise there is no need to do an edge-splitting operation at s , i.e., $|\delta_{K_{2C}}(s)| = 4C$). This implies that we can always choose an edge-splitting pair such that at least $2C$ edges between s and t are preserved.

Let z be the edge vector associated with the resulting graph after edge-splitting operations, i.e., $z_{(u,v)}$ equals the number of edges between u and v in the resulting graph. Furthermore, let $z' = z/2C$. Then $z'(\delta(S)) \geq 2$ for each cut S , $z'(\delta(v)) = 2$ for each vertex v , and $z'_{(s,t)} \geq 1$. Consider two different vertices u, v . We know $z'(\delta(u)) = z'(\delta(v)) = 2$ and $z'(\delta(\{u, v\})) \geq 2$. This implies $z'_{(u,v)} \leq 1$. In particular, $z'_{(s,t)} = 1$. Construct x' from z' as follows:

$$x'_e = \begin{cases} z'_e - 1 = 0, & \text{if } e = (s, t), \\ z'_e, & \text{otherwise.} \end{cases}$$

By the properties obtained for z' , we have x' is a feasible solution of **LP3**. Note that the edge-splitting operations never increase the total cost since the edge costs are metric on G . Therefore, the cost of x' is at most $c^H(x)$. In particular, if x is integral, we can set $C = 1$ in the procedure. In this case, x' is half-integral. \square

Conversely, any feasible solution of **LP3** can be transformed to a feasible solution of **LP6**: the idea is to replace each edge (u, v) in $E(G)$ by a shortest u - v path in H . Note that every solution of **LP3** is a feasible solution of the spanning tree polytope. Hence, it can be seen that the transformed solution is feasible for **LP6**, and, in particular, it satisfies the partition constraints in **LP6**. Hence,

$$\text{Opt}(\text{LP6}) \leq \text{Opt}(\text{LP3}), \quad \text{Opt}_{int}(\text{LP6}) \leq \text{Opt}_{int}(\text{LP3}). \quad (4.7)$$

By Lemma 4.3.2, we have the following result.

Corollary 4.3.3 $\text{Opt}(\text{LP6}) = \text{Opt}(\text{LP3})$.

However, **LP3** and **LP6** may differ in terms of the integral optimal value. Consider the graph with unit edge costs in Figure 4.1; this is meant to be the original graph H in the instance of the s - t path TSP.

Note that **LP6** is defined on the original graph but **LP3** is defined on the metric completion. Let ℓ be the length of the middle path in Figure 4.1. It is not hard to see that $\text{Opt}_{int}(\text{LP3}) \approx 3\ell$ but $\text{Opt}_{int}(\text{LP6}) \approx 2\ell$ when ℓ is sufficiently large. (For **LP6**, consider the integral solution with value 1 for every edge of the original graph.) In this case, $\frac{\text{Opt}_{int}(\text{LP3})}{\text{Opt}_{int}(\text{LP6})} \approx \frac{3}{2}$. Interestingly, $\frac{3}{2}$ can be proved to be an upper bound for this ratio. This example shows that the upper bound of $\frac{3}{2}$ is tight. To prove this upper bound, we present an algorithm to round a half-integral solution of **LP3** to an integral one by increasing the cost by a factor of at most $\frac{3}{2}$.

Apply the randomized Christofides' algorithm to a half-integral solution x of **LP3**. Let J be the random spanning tree obtained from x . Let F be a minimum-cost T -join for the set of wrong degree vertices T of J .

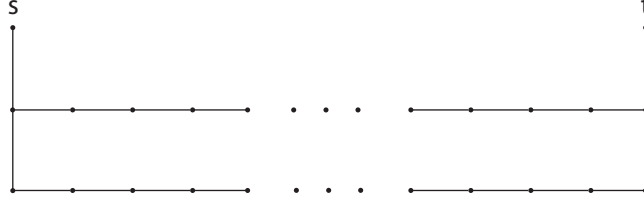


Figure 4.1: Tight example

Lemma 4.3.4 $x(\delta(S)) \geq 2$ for any T -odd cut S .

Proof. For any vertex $v \in V$, $x(\delta(v))$ is integral by the constraints of LP3. Since x_e is half-integral, $x(\delta(S)) = \sum_{v \in S} x(\delta(v)) - 2x(E(S))$ implies that $x(\delta(S))$ is integral. Suppose $x(\delta(S)) < 2$ for some T -odd cut S . Then we have $x(\delta(S)) = 1$. By the constraints of LP3, S must be an s - t cut. Note that $\mathbb{E}(\mathcal{X}^J) = x$ and $|J \cap \delta(S)| \geq 1$ since J is a random spanning tree. This implies $|J \cap \delta(S)| = 1$ always holds. However, since S is an s - t cut and also a T -odd cut, we have $|\delta(S) \cap J|$ is even by Lemma 3.1.2. This is a contradiction. \square

Theorem 4.3.5 If the input is a half-integral solution x of LP3, then the randomized Christofides' algorithm outputs a Hamiltonian s - t path with cost at most $\frac{3}{2}c(x)$.

Proof. By Lemma 4.3.4, $\frac{1}{2}x$ is a feasible solution of the T -join polyhedron LP5. This means $\mathbb{E}(c(F)) \leq \frac{1}{2}c(x)$. Therefore $\mathbb{E}(c(J)) + \mathbb{E}(c(F)) \leq \frac{3}{2}c(x)$. \square

Now we are ready to prove the ratio for the optimal integral values of the two LPs.

Theorem 4.3.6 $Opt_{int}(LP6) \leq Opt_{int}(LP3) \leq \frac{3}{2}Opt_{int}(LP6)$. Moreover, the bounds are tight.

Proof. The lower bound is due to (4.7). Now consider the upper bound. Let x be an optimal integral solution of LP6. By Lemma 4.3.2, there exists a half-integral solution x' of LP3 such that $c(x') \leq c^H(x)$. By Theorem 4.3.5, we can get an s - t Hamiltonian path with cost at most $\frac{3}{2}c(x')$. This means $Opt_{int}(LP3) \leq \frac{3}{2}c(x') \leq \frac{3}{2}c^H(x) = \frac{3}{2}Opt_{int}(LP6)$.

The tight example for the upper bound is shown in Figure 4.1. For the tightness of the lower bound, consider the graph H consisting of one path connecting s and t where every edge has unit cost. \square

4.4 Counterexample to two approaches

For the s - t path TSP, the main question is whether there exists a $\frac{3}{2}$ -approximation algorithm. When addressing this problem, two natural questions arise:

- Chapter 3 presented a simple $\frac{3}{2}$ -approximation algorithm for the s - t path TSP in the graphic case. Does it extend to give the same approximation factor for the general metric case?
- Does every spanning tree in a given convex decomposition of an optimal solution x of LP3 achieve a $\frac{3}{2}$ -approximation factor by adding a minimum-cost T -join to fix the wrong degree vertices?

The first question concerns the extension of the algorithm for the graphic case. The second question focuses on the role of randomness and probabilistic methods in the analysis of the recent LP-based approximation algorithms. We answer these questions negatively by providing a counterexample. In the following, we make the questions more precise and then show how our counterexample serves as a negative answer.

Recall from Chapter 3 that $H = (V, E(H))$ is a connected graph with unit edge costs c^H and G is the metric completion of H with metric costs c . Also recall some notation from Chapter 3: optimal solution x^* of LP1 (the graphic version of LP6), narrow cuts $\{Q_i\}_{1 \leq i \leq k}$ (for the solution x^* of LP1) and its corresponding vertex partition $\{L_i\}_{0 \leq i \leq k+1}$.

The algorithm in Chapter 3 constructs a minimal spanning tree on each L_i and then connects them together by a unit cost edge between each two consecutive L_i and L_{i+1} . This results in a spanning tree on H , which is called a *good spanning tree*. Then a minimum-cost T -join F_{good} is added to correct the wrong degree vertices of the good spanning tree. Since every edge in H has unit cost, the good spanning tree has minimum cost, which is at most $Opt(LP6)$ on H with unit edge costs. Furthermore, it is shown that the minimum-cost T -join F_{good} has cost at most $\frac{1}{2}Opt(LP6)$. This gives a $\frac{3}{2}$ -approximation factor in total.

The only part in the analysis using the graphic property is that the good spanning tree has cost at most $Opt(LP6)$. A natural extension of the definition of a good spanning tree would be as follows:

- In the general metric case, a good spanning tree is constructed by connecting the minimum-cost spanning tree in each L_i with a minimum-cost edge from L_i to L_{i+1} .

If the cost of this “extended” good spanning tree is bounded above by $Opt(LP6)$ in the general metric case, then it gives us a $\frac{3}{2}$ -approximation factor for the s - t path TSP. Unfortunately, this is

not true. To show this, we present our counterexample, a complete graph $G = H = H_b$ with metric edge costs c^{H_b} and vertex set $\{0, 1, \dots, 7\}$ where $s = 0, t = 7$. The metric edge costs c^{H_b} are given by the metric completion of the costs indicated in Figure 4.2 below. Note that for every edge e in Figure 4.2, $c_e^{H_b}$ is exactly the edge cost value shown in that figure.

Figure 4.2 shows the support graph of a feasible solution x^{H_b} of LP6, where the first number on each edge denotes the x^{H_b} value and the second number denotes the cost of the edge.

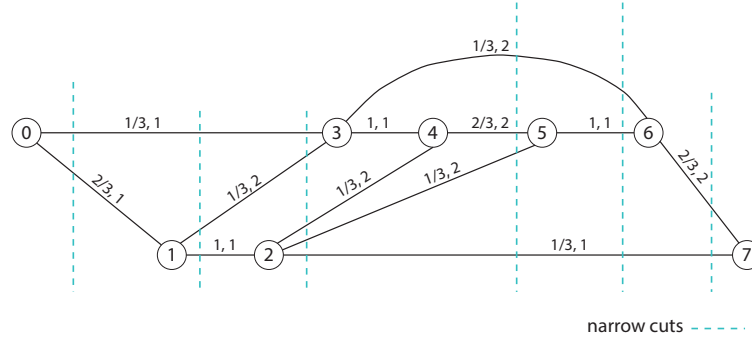


Figure 4.2: Support graph of x^{H_b} with edge x^{H_b} values and edge costs

Lemma 4.4.1 x^{H_b} is an optimal solution for LP6 with respect to c^{H_b} . Furthermore, x^{H_b} is an extreme point of the polyhedron of LP6 on H_b .

Proof. To show the optimality of x^{H_b} for LP6, it is sufficient to prove that x^{H_b} is an optimal solution of LP3 by Corollary 4.3.3. We use complementary slackness conditions to prove the optimality of x^{H_b} for LP3. Let \mathcal{S}_1 be the set of all s - t cuts and \mathcal{S}_2 be the set of all $\{s, t\}$ -even cuts. Let $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$.

(Dual of LP3)

$$\text{maximize : } y_s + y_t + 2 \sum_{v \notin \{s, t\}} y_v + \sum_{S \in \mathcal{S}_1} d_S + 2 \sum_{S \in \mathcal{S}_2} d_S - \sum_e u_e$$

subject to :

$$y_w + y_v - u_{(w, v)} + \sum_{(w, v) \in \delta(S), S \in \mathcal{S}} d_S \leq c_{(w, v)}, \quad (w, v) \in E$$

$$u, d \geq 0$$

The following dual solution y, d, u witnesses the optimality of x^{H_b} to **LP3** by the complementary slackness conditions:

- $u_{(1,2)} = u_{(3,4)} = \frac{2}{3}$, $u_{(5,6)} = \frac{4}{3}$, and $u_e = 0$ for any other edge e
- $d_{\{3,4,5,6\}} = \frac{1}{3}$ and $d_S = 0$ for any other S
- $y_0 = 0, y_2 = y_3 = \frac{2}{3}, y_1 = y_4 = y_5 = 1, y_6 = \frac{4}{3}, y_7 = \frac{1}{3}$

Hence x^{H_b} is also an optimal solution of **LP6**.

Denote the polyhedron of **LP6** on H_b by K . We now show that x^{H_b} is an extreme point of K . Otherwise, there exists $x^{H_b} \neq z \in K$ and $z' \in K$ such that $x^{H_b} = \lambda z + (1 - \lambda)z'$ for some $0 < \lambda < 1$.

Clearly, for any edge e not in the support graph of x^{H_b} , we have $z_e = 0$ by Lemma 4.1.2. We also apply Lemma 4.1.2 to $\delta(v)$ for each vertex v , and the cuts $S_1 = \{3, 4\}, S_2 = \{1, 2\}, S_3 = \{5, 6\}, S_4 = \{3, 4, 5, 6\}$. Then, $z(\delta(v)) = 1$ for $v = 0, 7$ and $z(\delta(v)) = 2$ for other vertices, and $z(\delta(S_j)) = 2$ for $1 \leq j \leq 4$. Hence, $z_e = 1$ for each $e \in E_1 = \{(3, 4), (1, 2), (5, 6)\}$. Let $a = z_{(0,3)}, b = z_{(4,5)}$. By the z -values on the edges in E_1 and the values $z(\delta(v))$ for $v \in V(H_b)$, we have $z_{(0,1)} = 1 - a, z_{(1,3)} = a, z_{(3,6)} = 1 - 2a, z_{(6,7)} = 2a, z_{(2,7)} = 1 - 2a, z_{(2,5)} = 1 - b, z_{(2,4)} = 1 - b$. Now consider $\delta(2)$ and $\delta(S_4)$. Then

$$2(1 - b) + (1 - 2a) + 1 = 2, \quad 4a + 2(1 - b) = 2.$$

Hence, $a = \frac{1}{3}, b = \frac{2}{3}$. By checking each edge, $z = x^{H_b}$. This is a contradiction. Therefore, x^{H_b} is an extreme point of K . \square

Note that the analysis in the proof of Lemma 4.4.1 also shows that x^{H_b} is an extreme point of the polytope of **LP3** on H_b .

The cost of the corresponding good spanning tree is 10 and is shown in Figure 4.3. The number on the edge between 3 and 4 in Figure 4.3 is the edge cost. The numbers below the dashed narrow cuts are the minimum costs of the edges crossing the narrow cuts to connect two consecutive parts. By Lemma 4.4.1, we know the optimal value of **LP6** is $c^{H_b}(x^{H_b}) = 9\frac{2}{3}$. So, we can see that the cost of the good spanning tree is strictly larger than the optimal value of **LP6**. This refutes the statement that the cost of the “extended” good spanning tree can be upper bounded by $Opt(\text{LP6})$.

Interestingly, this instance also illustrates that probabilistic methods are important for the analyses of improved LP-based approximation algorithms such as the “randomized Christofides’ algorithm” or its deterministic version the “best-of-many Christofides’ algorithm” (see [1]). The

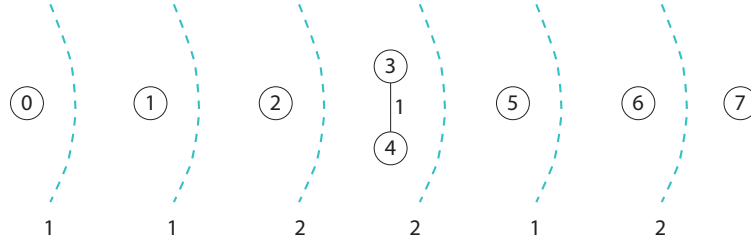


Figure 4.3: Cost of the good spanning tree

randomized Christofides' algorithm obtains a better approximation factor by sampling a spanning tree J from the convex decomposition of x^* . However, is it true that for an arbitrary spanning tree in the support of a given convex decomposition, the cost of the spanning tree plus a minimum-cost T -join is at most $\frac{3}{2}Opt(\text{LP3})$? In the rest of this section, via the instance H_b , we show this statement is false in general.

We recall the optimal solution x^{H_b} of **LP3** on H_b with metric costs c^{H_b} . We know that x^{H_b} is in the spanning tree polytope **LP4**. The tight constraints of x^{H_b} for the inequality constraints of **LP4** are illustrated as dashed circles in the Figure 4.4 except the tight constraints for $V - \{s\}$, $V - \{t\}$, $V - \{s, t\}$.

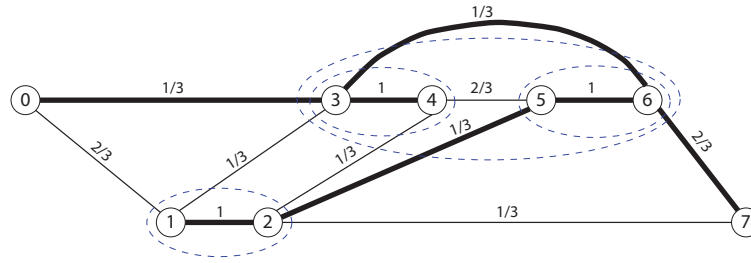


Figure 4.4: Tree J_b

By Lemma 4.1.2, the tree J_b with the dark edges in the graph of Figure 4.4 is in some convex decomposition of x^{H_b} in LP4, i.e., J_b is a spanning tree in the support of some convex decomposition of x^{H_b} . Let T_b be the set of wrong degree vertices of J_b , i.e., $T_b = \{1, 3, 4, 6\}$. $F_b = \{(3, 6), (1, 4)\}$ is a minimum-cost T_b -join with cost 5. Hence, the total cost of the disjoint union of J_b and F_b is 15, which is larger than $\frac{3}{2}$ times the optimal value $c^{H_b}(x^{H_b}) = 9\frac{2}{3}$ of LP3. This shows the importance of the probabilistic techniques in the analysis of the “randomized Christofides’ algorithm” or its deterministic version the “best-of-many Christofides’ algorithm”. Note that the minimum-cost T_b -join F_b to fix the wrong degree vertices of J_b is also larger than half of the optimal value $9\frac{2}{3}$ of LP3.

Chapter 5

On Integrality Ratios for Asymmetric TSP in the Sherali-Adams System

The Traveling Salesman Problem is to find a minimum-cost tour of a set of cities; the tour should visit each city exactly once. The most well known version of this problem is the symmetric one (i.e., TSP), where the distance (a.k.a. cost) from city i to city j is equal to the distance (cost) from city j to city i . The more general version is called Asymmetric TSP (ATSP), and it does not have the symmetry restriction on the costs. Throughout this chapter,¹ we assume that the costs satisfy the triangle inequalities, i.e., the costs are metric.

Linear programming relaxations play a central role in solving TSP or ATSP, both in practice and in the theoretical setting of approximation algorithms. The most well known relaxation (and the one that is most useful for theory and practice) is due to Dantzig, Fulkerson and Johnson; we call it the standard LP relaxation or the DFJ LP relaxation (see Section 2.1). There is a further relaxation of the standard LP relaxation that is of interest; we call it the balanced LP relaxation (see Section 2.1); it is obtained from the standard LP relaxation by replacing the indegree and outdegree constraint at each vertex by a balance (equation) constraint. We may denote the balance LP relaxation by Bal LP for short. For metric costs, the optimal value of the standard LP relaxation is the same as the optimal value of the balanced LP relaxation; this is a well known fact, see [57], [10, Footnote 3].

For both TSP and ATSP, significant research efforts have been devoted over several decades to prove bounds on the integrality ratio of the standard LP (DFJ LP) relaxation. For TSP, methods based on Christofides' algorithm show that the integrality ratio is $\leq \frac{3}{2}$, whereas the best lower

¹ This chapter is based on joint work with Joseph Cheriyan, Konstantinos Georgiou, and Sahil Singla. The results appear in the submitted manuscript [15], an extended abstract of which has been published [14].

bound known on the integrality ratio is $\frac{4}{3}$. Closing this gap is a major open problem in the area. For ATSP, a result of Asadpour et al. [5] showed that the integrality ratio is $\leq O(\log n / \log \log n)$. Very recently, Anari et al. [3] improved the upper bound on the integrality ratio to $\text{polyloglog}(n)$ for ATSP. On the other hand, Charikar et al. [10] showed a lower bound of 2 on the integrality ratio, thereby refuting an earlier conjecture of Carr and Vempala [9] that the integrality ratio is $\leq \frac{4}{3}$.

Lampis [45] and Papadimitriou and Vempala [56], respectively, have proved hardness-of-approximation thresholds of $\frac{185}{184}$ for TSP and $\frac{117}{116}$ for ATSP; both results assume that $P \neq NP$. Karpinski et al. [42] have improved both hardness-of-approximation thresholds to 123/122 and 75/74, respectively, assuming that $P \neq NP$.

Our goal is to prove lower bounds on the integrality ratios for the tighter LP relaxations for ATSP obtained by applying the Sherali-Adams system.

Starting with the work of Stephen and Tunçel [64] and Arora et al. [4], substantial research efforts have been devoted to showing that tightened relaxations (for many levels) fail to reduce the integrality ratio for many combinatorial optimization problems (see [19] for a list of negative results). A key paper by Fernández de la Vega and Kenyon-Mathieu [24] introduced a probabilistic interpretation of the SA system, and based on this, negative results (for the SA system) have been proved for a number of combinatorial problems; also see Charikar et al. [11], and Benabbas et al. [7]. At the moment, it is not clear that methods based on [24] could give negative results for TSP and its variants, because the natural LP relaxations (of TSP and related problems) have “global constraints.”

To the best of our knowledge, there are only two previous papers with negative results for Lift-and-Project systems applied to TSP and its variants. Cheung [18] proves a lower bound of $\frac{4}{3}$ on the integrality ratio for TSP, for $O(1)$ levels of the SDP version of Lovász-Schrijver system. For ATSP, Watson [68] proves an integrality ratio of $\frac{3}{2}$ for level 1 of the Lovász-Schrijver system, starting from the balanced LP relaxation (in fact, both the systems LS and SA give the same relaxation at level one).

5.1 Our results

Our main result is a generic construction of fractional feasible solutions for any level t of the SA system starting from the standard LP (DFJ LP) relaxation of ATSP. We have a similar but considerably simpler construction when the starting LP for the SA system is the balanced LP relaxation. Our results on integrality ratios are direct corollaries.

We have the following results pertaining to the balanced LP relaxation of ATSP: We formulate a property of digraphs that we call the good decomposition property, and given any digraph with this property, we construct a vector y on the edges such that y is a fractional feasible solution to the level- t tightening of the balanced LP relaxation by the Sherali-Adams system. Charikar, Goemans, and Karloff (CGK) [10] constructed a family of digraphs for which the balanced LP relaxation has an integrality ratio of 2. We show that the digraphs in the CGK family have the good decomposition property, hence, we obtain an integrality ratio for level t of SA. In more detail, we prove that for any integer $t \geq 0$ and small enough $\epsilon > 0$, there is a digraph G from the CGK family on $\nu = \nu(t, \epsilon) = O((t/\epsilon)^{t/\epsilon})$ vertices such that the integrality ratio of the level- t tightening of Bal LP is at least $1 + \frac{1-\epsilon}{t+1} \approx 2, \frac{3}{2}, \frac{4}{3}, \frac{5}{4}, \dots$ (where $t = 0$ identifies the original relaxation).

Our main result pertains to the standard LP relaxation of ATSP. Our key contribution is to identify a structural property of digraphs that allows us to construct fractional feasible solutions for the level- t tightening of the standard LP relaxation by the Sherali-Adams system. This construction is much more difficult than the construction for the balanced LP relaxation. We present a simple family of digraphs that satisfy the structural property, and this immediately gives our results on integrality ratios. We prove that for any integer $t \geq 0$ and small enough $\epsilon > 0$, there are digraphs G on $\nu = \nu(t, \epsilon) = O(t/\epsilon)$ vertices such that the integrality ratio of the level- t tightening of the standard LP relaxation on G is at least $1 + \frac{1-\epsilon}{2t+3} \approx \frac{4}{3}, \frac{6}{5}, \frac{8}{7}, \frac{10}{9}, \dots$. The *rank* of a starting relaxation (or polytope) is defined to be the minimum number of tightenings required to find the integral hull (in the worst case). An immediate corollary is that the SA-rank of the standard LP relaxation on a digraph $G = (V, E)$ is at least linear in $|V|$, whereas, the rank in terms of the number of edges is $\Omega(\sqrt{|E|})$ (since the LP is on a complete digraph, namely, the metric completion).

Our results for the balanced LP relaxation and for the standard LP relaxation are incomparable, because the SA system starting from the standard LP relaxation is strictly stronger than the SA system starting from the balanced LP relaxation, although both the level zero LPs have the same optimal value, assuming metric costs. (In fact, there is an example on 5 vertices [27, Figure 4.4, p.60] such that the optimal values of the level-1 tightenings are different: $9\frac{1}{3}$ for the balanced LP relaxation and 10 for the standard LP relaxation.)

Finally, we extend our main results to the natural relaxation of path ATSP (minimum-cost Hamiltonian dipath from a given source vertex to a given sink vertex), and we obtain integrality ratios $\geq 1 + \frac{2-\epsilon}{3t+4} \approx \frac{3}{2}, \frac{9}{7}, \frac{6}{5}, \frac{15}{13}, \dots$ for the level- t SA tightenings. Our result on path ATSP is obtained by “reducing” from the result for ATSP; the idea behind this comes from an analogous result of Watson [68] in the symmetric setting; Watson gives a method for transforming Cheung’s [18] result on the integrality ratio for TSP to obtain a lower bound on the integrality ratio for path TSP.

5.2 Preliminaries

When discussing a digraph (directed graph), we use the terms dicycle (directed cycle), etc., but we use the term edge rather than directed edge or arc.

Consider a strongly connected digraph $G = (V, E)$ with nonnegative edge costs $c \in \mathbb{R}^E$. By the *metric completion* of G , we mean the complete digraph G' on V with the edge costs c' , where $c'(v, w)$ is taken to be the minimum cost (w.r.t. c) of a v, w dipath of G .

An *Eulerian subdigraph* of G is defined as follows: the vertex set is V and the edge set is a “multi-subset” of E (that is, each edge in E occurs zero or more times) such that (i) the indegree of every vertex equals its outdegree, and (ii) the subdigraph is weakly connected (i.e., the underlying undirected graph is connected). The ATSP on the metric completion G' of G is equivalent to finding a minimum-cost Eulerian subdigraph of G .

For a positive integer t and a ground set U , we let \mathcal{P}_t denote the family of subsets of U of size at most t , i.e., $\mathcal{P}_t = \{S : S \subseteq U, |S| \leq t\}$ (see Section 2.2). We usually take the ground set to be the set of edges of a fixed digraph. Now, let G be a digraph, and let the ground set (for \mathcal{P}_t) be $E = E(G)$. Let E' be a subset of E . Let $\mathbf{1}^{E', t}$ denote a vector indexed by elements of \mathcal{P}_t such that for any $S \in \mathcal{P}_t$, $\mathbf{1}_S^{E', t} = 1$ if $S \subseteq E'$, and $\mathbf{1}_S^{E', t} = 0$, otherwise. Note that $\mathbf{1}^{E', 1}$ has the entry for \emptyset at 1, and the other entries give the incidence vector of E' .

5.2.1 LP relaxations for Asymmetric TSP

Let $G = (V, E)$ be a digraph with nonnegative edge costs c . Let $\widehat{\text{ATSP}}_{DFJ}(G)$ be the feasible region (polytope) of the following linear program that has a variable x_e for each edge e of G :

$$\begin{array}{ll}
 \text{minimize: } \sum_{e \in E} c_e x_e & \\
 \text{subject to: } x(\delta^{\text{in}}(S)) \geq 1 & \forall \emptyset \subsetneq S \subsetneq V \\
 x(\delta^{\text{out}}(S)) \geq 1 & \forall \emptyset \subsetneq S \subsetneq V \\
 x(\delta^{\text{in}}(v)) = 1, x(\delta^{\text{out}}(v)) = 1 & \forall v \in V \\
 0 \leq x_e \leq 1 & \forall e \in E
 \end{array}$$

In particular, when G is a complete digraph with metric costs, the above linear program is the standard LP (DFJ LP) relaxation of ATSP (see Section 2.1).

We obtain the balanced LP relaxation (Bal LP) from the standard LP relaxation by replacing the two constraints $x(\delta^{in}(v)) = 1$, $x(\delta^{out}(v)) = 1$ by the constraint $x(\delta^{in}(v)) = x(\delta^{out}(v))$, for each vertex v . Let $\widehat{\text{ATSP}}_{BAL}(G)$ be the feasible region (polytope) of Bal LP.

$$\begin{array}{ll}
\text{minimize: } \sum_{e \in E} c_e x_e & \\
\text{subject to: } x(\delta^{in}(S)) \geq 1 & \forall \emptyset \subsetneq S \subsetneq V \\
x(\delta^{out}(S)) \geq 1 & \forall \emptyset \subsetneq S \subsetneq V \\
x(\delta^{in}(v)) = x(\delta^{out}(v)) & \forall v \in V \\
0 \leq x_e \leq 1 & \forall e \in E
\end{array}$$

In particular, when G is a complete digraph with metric costs, the above linear program is the balanced LP relaxation of ATSP (see Section 2.1).

Our construction of fractional feasible solutions exploits the structure of the original digraph. This is the reason for discussing the polytopes on the original digraph (and not only on the complete digraph). To justify this, we observe that any feasible solution for the original digraph can be extended to a feasible solution for the complete digraph by “padding with zeros.” (This argument is formalized in Section 5.2.2).

5.2.2 The Sherali-Adams system

In this section, we recall the definition of the Sherali-Adams system from Section 2.2 and present some basic properties. Here, we use the linearized definition of the Sherali-Adams system.

Definition 5.2.1 (The Sherali-Adams system) Consider a polytope $\widehat{P} \subseteq [0, 1]^n$ over the variables y_1, \dots, y_n , and its description by a system of linear constraints of the form $\sum_{i=1}^n a_i y_i \geq b$; note that the constraints $y_i \geq 0$ and $-y_i \geq -1$ for all $i \in \{1, \dots, n\}$ are included in the system. The level- t **Sherali-Adams tightened relaxation** $\text{SA}^t(\widehat{P})$ of \widehat{P} , is an LP over the variables $\{y_S : S \subseteq \{1, 2, \dots, n\}, |S| \leq t+1\}$ (thus, $y \in \mathbb{R}^{\mathcal{P}_{t+1}}$ where \mathcal{P}_{t+1} has ground set $\{1, 2, \dots, n\}$); moreover, we have $y_\emptyset = 1$. For every constraint $\sum_{i=1}^n a_i y_i \geq b$ of \widehat{P} and for every disjoint $S, Q \subseteq \{1, \dots, n\}$ with $|S| + |Q| \leq t$, the following is a constraint of the level- t

Sherali-Adams relaxation.

$$\sum_{i=1}^n a_i \sum_{\emptyset \subseteq T \subseteq Q} (-1)^{|T|} y_{S \cup T \cup \{i\}} \geq b \sum_{\emptyset \subseteq T \subseteq Q} (-1)^{|T|} y_{S \cup T}. \quad (5.1)$$

We will use a convenient abbreviation:

$$z_{S,Q} := \sum_{\emptyset \subseteq T \subseteq Q} (-1)^{|T|} y_{S \cup T},$$

where $z_{S,Q}$ are auxiliary variables between 0 and 1.

Informally speaking, the level- t Sherali-Adams relaxation is derived by multiplying any constraint of the original relaxation by the high degree polynomial

$$\prod_{j \in S} y_j \prod_{j \in Q} (1 - y_j),$$

where S, Q are disjoint subsets of $\{1, \dots, n\}$ with $|S| + |Q| \leq t$. After expanding the products, we obtain a polynomial of degree at most $t+1$. Replacing any occurrences of $\prod_{i \in S} y_i$ by the corresponding variable y_S for all $S \subseteq \{1, \dots, n\}$ gives the constraint described in Inequality (5.1) (Definition 5.2.1).

There are a number of approaches for certifying that $y \in \text{SA}^t(\hat{P})$ for a given y . One popular approach is to give a probabilistic interpretation to the entries of y , satisfying certain conditions. We follow an alternative approach, that is standard, see [47], [65, Lemma 2.9], but has been rarely used in the context of integrality ratios.

First, we recall some notation from Section 2.2. Given a polytope $\hat{P} \subseteq [0, 1]^n$, consider the cone $P = \{y_\emptyset(1, y) : y_\emptyset \geq 0, y \in \hat{P}\}$. (Throughout this chapter, we use an accented symbol to denote a polytope, e.g., \hat{P} , and the symbol (without accent) to denote the associated cone, e.g., P .) Recall that $\text{SA}^t(\hat{P})$ is a polytope whereas $\text{SA}^t(P)$ is a cone. Also, note that $\text{SA}^t(\hat{P}) = \{y : y_\emptyset = 1, y \in \text{SA}^t(P)\}$.

For a vector y indexed by subsets of $\{1, \dots, n\}$ of size at most $t+1$, define a *shift operator* “ $*$ ” as follows: for every $e \in \{1, \dots, n\}$, let $e * y$ to be a vector indexed by subsets of $\{1, \dots, n\}$ of size at most t , such that $(e * y)_S := y_{S+e}$. We have the following fact, [65, Lemma 2.9].

Fact 5.2.2 . $y \in \text{SA}^t(P)$ *if and only if* $e * y \in \text{SA}^{t-1}(P)$, *and* $y - e * y \in \text{SA}^{t-1}(P)$, $\forall e \in \{1, \dots, n\}$.

The reader familiar with the Lovász-Schrijver system may recognize the similarity of its definition with the characterization of the Sherali-Adams system of Fact 5.2.2 (see Section 2.2). In fact, the SA system differs from the LS system only in that it imposes additional consistency conditions; namely, the moment vector y , indexed by subsets of size $t + 1$, has to be fixed beforehand. This seemingly small detail gives the SA system enhanced power compared to the LS system.

Eliminating Variables to 0

In our discussion of the standard LP relaxation and the balanced LP relaxation, it will be convenient to restrict the support to the edge set of a given digraph rather than the complete digraph. Thus, we assume that some of the variables are absent. Formally, this is equivalent to setting these variables in advance to zero. As long as the nonzero variables induce a feasible solution, we are justified in setting the other variables to zero. The following result formalizes the arguments.

Proposition 5.2.3 *Let \hat{P} be the feasible region (polytope) of a linear program. Let C be a set of indices (of the variables) that does not contain the support of any “positive constraint” of \hat{P} , where a constraint $\sum_{i=1}^n a_i y_i \geq b$ of \hat{P} is called positive if $b > 0$. Let \hat{P}_C be the feasible region (polytope) of the linear program obtained by removing all variables with indices in C from the constraints of the linear program of \hat{P} (informally, the new LP fixes all variables with indices in C at zero). Then, for the SA system, for any feasible solution y to the level- t tightening of \hat{P}_C , there exists a feasible solution y' to the level- t tightening of \hat{P} ; moreover, y' is obtained from y by fixing variables, indexed by subsets intersecting C , to zero.*

Proof. For $y \in \text{SA}^t(\hat{P}_C)$, the “extension” y' of y is defined as follows:

$$y'_S = \begin{cases} y_S & , \text{ if } S \cap C = \emptyset \\ 0 & , \text{ otherwise} \end{cases}$$

For the corresponding auxiliary variables z , this would imply that

$$z'_{S,Q} = \begin{cases} 0 & , \text{ if } S \cap C \neq \emptyset \\ z_{S,Q-C} & , \text{ otherwise} . \end{cases}$$

In order to show that $y' \in \text{SA}^t(\hat{P})$, we need to verify that for every pair of sets S, Q as in Definition 5.2.1, we have $\sum_{i=1}^n a_i z'_{S \cup \{i\}, Q} \geq b z'_{S,Q}$.

First we note that if $S \cap C \neq \emptyset$, then for every i we have $z'_{S \cup \{i\}, Q} = z'_{S, Q} = 0$, and hence the constraint is satisfied trivially.

For the remaining case $S \cap C = \emptyset$, we have

$$\begin{aligned}
\sum_{i=1}^n a_i z'_{S \cup \{i\}, Q} &= \sum_{i \in C} a_i z'_{S \cup \{i\}, Q} + \sum_{i \notin C} a_i z'_{S \cup \{i\}, Q} \\
&= \sum_{i \notin C} a_i z'_{S \cup \{i\}, Q} \\
&= \sum_{i \notin C} a_i z_{S \cup \{i\}, Q-C} \\
&\geq b z_{S, Q-C} \\
&= b z'_{S, Q-C} \\
&= b z'_{S, Q},
\end{aligned} \tag{5.2}$$

where (5.2) follows from the validity of the corresponding constraint of \hat{P}_C ; here, we use the fact that C does not contain the support of any positive constraint – otherwise, the summation $\sum_{i \notin C} (\dots)$ would be zero since the index set $\{i : i \notin C\}$ would be empty, and hence, the inequality $0 = \sum_{i \notin C} (\dots) \geq b z_{S, Q-C}$ would fail to hold for $b > 0$ and $z_{S, Q-C} > 0$. \square

5.3 SA applied to the Balanced LP relaxation of ATSP

5.3.1 Certifying a feasible solution

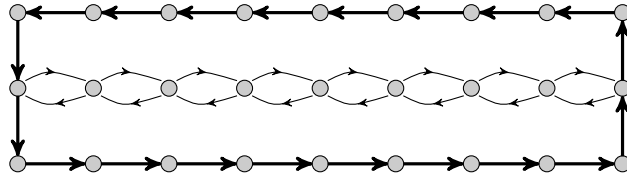


Figure 5.1: A digraph G with a good decomposition given by the dicycle with thick edges, and the length 2 dicycles C_j formed by the anti-parallel pairs of thin edges; $G - E(C_j)$ is strongly connected for each dicycle C_j .

A strongly connected digraph $G = (V, E)$ is said to have a *good decomposition* with *witness set* \mathcal{F} if the following hold

- (i) E partitions into edge-disjoint dicycles C_1, C_2, \dots, C_N , that is, there exist edge-disjoint dicycles C_1, C_2, \dots, C_N such that $E = \bigcup_{1 \leq j \leq N} E(C_j)$; let \mathcal{N} denote the set of indices of these dicycles, thus $\mathcal{N} = \{1, \dots, N\}$;
- (ii) moreover, there exists a nonempty subset \mathcal{F} of \mathcal{N} such that for each $j \in \mathcal{F}$ the digraph $G - E(C_j)$ is strongly connected.

Let $\overline{\mathcal{F}}$ denote $\mathcal{N} - \mathcal{F}$. For an edge e , we use $\text{index}(e)$ to denote the index j of the dicycle $C_j, j \in \mathcal{N}$ that contains e . In this section, by a dicycle C_i, C_j , etc., we mean one of the dicycles C_1, \dots, C_N , and we identify a dicycle C_j with its edge set, $E(C_j)$. See Figure 5.1 for an illustration of a good decomposition of a digraph.

Informally speaking, our plan is as follows: for digraph G that has a good decomposition with witness set \mathcal{F} , we construct a feasible solution to $\text{SA}^t(\widehat{\text{ATSP}}_{\text{BAL}}(G))$ by assigning the same fractional value to the edges of the dicycles C_j with $j \in \mathcal{F}$, while assigning the value 1 to the edges of the dicycles C_i with $i \in \overline{\mathcal{F}}$ (this is not completely correct; we will refine this plan). Let $\text{ATSP}_{\text{BAL}}(G)$ be the associated cone of $\widehat{\text{ATSP}}_{\text{BAL}}(G)$.

Definition 5.3.1 *Let t be a nonnegative integer. For any set $S \subseteq E$ of size $\leq t + 1$, and any subset \mathcal{I} of \mathcal{F} , let $F^{\mathcal{I}}(S)$ denote the set of indices $j \in \mathcal{F} - \mathcal{I}$ such that $E(C_j) \cap S \neq \emptyset$; moreover, let $f^{\mathcal{I}}(S)$ denote $|F^{\mathcal{I}}(S)|$, namely, the number of dicycles C_j with indices in $\mathcal{F} - \mathcal{I}$ that intersect S .*

Definition 5.3.2 *For a nonnegative integer t and for any subset \mathcal{I} of \mathcal{F} , let $y^{\mathcal{I}, t}$ be a vector indexed by the elements of \mathcal{P}_{t+1} and defined as follows:*

$$y_S^{\mathcal{I}, t} = \frac{t + 2 - f^{\mathcal{I}}(S)}{t + 2}, \quad \forall S \in \mathcal{P}_{t+1}$$

Theorem 5.3.3 *Let $G = (V, E)$ be a strongly connected digraph that has a good decomposition, and let \mathcal{F} be the witness set. Then*

$$y^{\mathcal{I}, t} \in \text{SA}^t(\widehat{\text{ATSP}}_{\text{BAL}}(G)), \quad \forall t \geq 0, \forall \mathcal{I} \subseteq \mathcal{F}.$$

In order to prove our integrality ratio result for $\widehat{\text{ATSP}}_{\text{BAL}}$, we will invoke Theorem 5.3.3 for $\mathcal{I} = \emptyset$ (the more general setting of the theorem is essential for our induction proof; we give a high-level explanation in the last paragraph of the proof of Theorem 5.3.3 below). Since also only the values of $y^{\emptyset, t}$ indexed at singleton edges affect the integrality ratio, it is worthwhile to summarize all relevant quantities in the next corollary.

Corollary 5.3.4 *We have*

$$y^{\emptyset, t} \in \text{SA}^t(\widehat{\text{ATSP}}_{\text{BAL}}(G)), \quad \forall t \geq 0.$$

Moreover, for each dicycle C_j , $j \in \mathcal{N}$, and each edge e of C_j we have

$$y_e^{\emptyset, t} = \begin{cases} \frac{t+1}{t+2}, & \text{if } j \in \mathcal{F} \\ 1, & \text{otherwise.} \end{cases} \quad (5.3)$$

Informally speaking, we assign the value 1 (rather than a fractional value) to the edges of the dicycles C_j with $j \in \mathcal{I} \subseteq \mathcal{F}$. For the sake of exposition, we call the dicycles C_j with $j \in \mathcal{F} - \mathcal{I}$ the *fractional dicycles*, and we call the remaining dicycles C_i (thus $i \in \mathcal{I} \cup \overline{\mathcal{F}}$) the *integral dicycles*.

Proof of Theorem 5.3.3: To prove Theorem 5.3.3, we need to prove

$$y^{\mathcal{I}, t} \in \text{SA}^t(\text{ATSP}_{\text{BAL}}(G)).$$

We prove this by induction on t .

Note that $y_{\emptyset}^{\mathcal{I}, t} = 1$ by Definition 5.3.2.

The induction basis is important, and it follows easily from the good decomposition property. In Lemma 5.3.8 (below) we show that $y^{\emptyset, 0} \in \text{SA}^0(\text{ATSP}_{\text{BAL}}(G))$. We conclude that $y^{\mathcal{I}, 0}$ satisfies the first two sets of constraints of $\text{ATSP}_{\text{BAL}}(G)$, since $y^{\mathcal{I}, 0} \geq y^{\emptyset, 0}$ (this follows from Definitions 5.3.1, 5.3.2, since $F^{\mathcal{I}}(S) \subseteq F^{\emptyset}(S)$). As for the balance constraints, it is enough to observe that every vertex of our instance (see Figure 5.1) is incident to pairs of outgoing and ingoing edges, which due to Definition 5.3.2 are assigned the same value. Finally, again by Definition 5.3.2, and for all edges e , we have $0 \leq y_e^{\mathcal{I}, 0} \leq 1$. All the above imply that $y^{\mathcal{I}, 0} \in \text{SA}^0(\text{ATSP}_{\text{BAL}}(G))$, $\forall \mathcal{I} \subseteq \mathcal{F}$, as wanted.

In the induction step, we assume that $y^{\mathcal{I}, t} \in \text{SA}^t(\text{ATSP}_{\text{BAL}}(G))$ for some integer $t \geq 0$ (the induction hypothesis), and we apply the recursive definition based on the shift operator, namely, $y^{\mathcal{I}, t+1} \in \text{SA}^{t+1}(\text{ATSP}_{\text{BAL}}(G))$ iff for each $e \in E$

$$e * y^{\mathcal{I}, t+1} \in \text{SA}^t(\text{ATSP}_{\text{BAL}}(G)), \quad (5.4)$$

$$y^{\mathcal{I}, t+1} - e * y^{\mathcal{I}, t+1} \in \text{SA}^t(\text{ATSP}_{\text{BAL}}(G)). \quad (5.5)$$

Lemma 5.3.6 (below) proves (5.4) and Lemma 5.3.7 (below) proves (5.5). \square

We prove that $e * y^{\mathcal{I}, t+1}$ is in $\text{SA}^t(\text{ATSP}_{\text{BAL}}(G))$ by showing that for some edges e , $e * y^{\mathcal{I}, t+1}$ is

a scalar multiple of $y^{\mathcal{I}', t}$, where $\mathcal{I}' \supsetneq \mathcal{I}$ (see Equation (5.6) in Lemma 5.3.6); thus, the induction hinges on the use of \mathcal{I} .

Before proving Lemma 5.3.6 and Lemma 5.3.7, we show that $y^{\mathcal{I}, t+1}$, restricted to \mathcal{P}_{t+1} , can be written as a convex combination of $y^{\mathcal{I}, t}$ and the integral feasible solution $\mathbf{1}^{E, t+1}$. This is used in the proof of Lemma 5.3.6; for some of the edges $e \in E$, we show that $e * y^{\mathcal{I}, t+1} = y^{\mathcal{I}, t+1}$ (see Equation (5.6)), and then we have to show that the latter is in $\text{SA}^t(\text{ATSP}_{\text{BAL}}(G))$.

Fact 5.3.5 . *Let t be a nonnegative integer and let \mathcal{I} be a subset of \mathcal{F} . Then for any $S \in \mathcal{P}_{t+1}$ we have $y_S^{\mathcal{I}, t+1} = \frac{t+2}{t+3}y_S^{\mathcal{I}, t} + \frac{1}{t+3}\mathbf{1}_S^{E, t+1}$.*

Proof. We have $S \subseteq E$, $|S| \leq t+1$, and we get $\mathbf{1}_S^{E, t+1} = 1$ from the definition. Thus,

$$y_S^{\mathcal{I}, t+1} = \frac{t+3 - f^{\mathcal{I}}(S)}{t+3} = \frac{t+2 - f^{\mathcal{I}}(S)}{t+3} + \frac{1}{t+3} = \frac{t+2}{t+3}y_S^{\mathcal{I}, t} + \frac{1}{t+3}\mathbf{1}_S^{E, t+1}.$$

□

Lemma 5.3.6 *Suppose that $y^{\mathcal{I}, t} \in \text{SA}^t(\text{ATSP}_{\text{BAL}}(G))$, for each $\mathcal{I} \subseteq \mathcal{F}$. Then for all $e \in E$ and for all $\mathcal{I} \subseteq \mathcal{F}$ we have $e * y^{\mathcal{I}, t+1} \in \text{SA}^t(\text{ATSP}_{\text{BAL}}(G))$*

Proof. For any $S \in \mathcal{P}_{t+1}$, the definition of the shift operator gives $(e * y^{\mathcal{I}, t+1})_S = y_{S+e}^{\mathcal{I}, t+1}$. Let $C(e)$ denote the dicycle containing edge e , and recall that $\text{index}(e)$ denotes the index of $C(e)$.

We first show that

$$e * y_S^{\mathcal{I}, t+1} = \begin{cases} \frac{t+2}{t+3}y_S^{\mathcal{I}+\text{index}(e), t} & \text{if } \text{index}(e) \in \mathcal{F} - \mathcal{I} \\ y_S^{\mathcal{I}, t+1} & \text{otherwise} \end{cases} \quad (5.6)$$

If $\text{index}(e) \in \mathcal{I} \cup \overline{\mathcal{F}}$, that is, the dicycle $C(e)$ is not “fractional,” then Definition 5.3.2 directly gives $y_{S+e}^{\mathcal{I}, t+1} = y_S^{\mathcal{I}, t+1}$. Otherwise, if $\text{index}(e) \in \mathcal{F} - \mathcal{I}$, then from Definition 5.3.2 we see that if $C(e) \cap S \neq \emptyset$, then $F^{\mathcal{I}}(S+e) = F^{\mathcal{I}}(S)$, and otherwise, $f^{\mathcal{I}}(S+e) = f^{\mathcal{I}}(S) + 1$. Hence,

$$(e * y^{\mathcal{I}, t+1})_S = \begin{cases} \frac{t+3-f^{\mathcal{I}}(S)}{t+3} & \text{if } C(e) \cap S \neq \emptyset \\ \frac{t+2-f^{\mathcal{I}}(S)}{t+3} & \text{if } C(e) \cap S = \emptyset \end{cases} \quad (5.7)$$

$$= \frac{t+2}{t+3}y_S^{\mathcal{I}+\text{index}(e), t} \quad (5.8)$$

where in the last line we use Definition 5.3.2 to infer that $f^{\mathcal{I}+index(e)}(S) = f^{\mathcal{I}}(S) - 1$, if $C(e) \cap S \neq \emptyset$, and $f^{\mathcal{I}+index(e)}(S) = f^{\mathcal{I}}(S)$, otherwise.

Note that Fact 5.3.5 along with $y^{\mathcal{I},t} \in SA^t(ATSP_{BAL}(G))$ implies that $y^{\mathcal{I},t+1}$, restricted to \mathcal{P}_{t+1} , is in $SA^t(ATSP_{BAL}(G))$ because it can be written as a convex combination of $y^{\mathcal{I},t}$ and an integral feasible solution $\mathbf{1}^{E,t+1}$. Equation (5.6) proves Lemma 5.3.6 because both $y^{\mathcal{I}+index(e),t}$ and $y^{\mathcal{I},t+1}$ (restricted to \mathcal{P}_{t+1}) are in $SA^t(ATSP_{BAL}(G))$. \square

Lemma 5.3.7 *Suppose that $y^{\mathcal{I},t} \in SA^t(ATSP_{BAL}(G))$, for each $\mathcal{I} \subseteq \mathcal{F}$. Then for all $e \in E$ and for all $\mathcal{I} \subseteq \mathcal{F}$ we have $y^{\mathcal{I},t+1} - e * y^{\mathcal{I},t+1} \in SA^t(ATSP_{BAL}(G))$.*

Proof. Let $C(e)$ denote the dicycle containing edge e , and recall that $index(e)$ denotes the index of $C(e)$. If $index(e) \in \mathcal{I} \cup \overline{\mathcal{F}}$, then we have $F^{\mathcal{I}}(S + e) = F^{\mathcal{I}}(S)$, $\forall S \in \mathcal{P}_{t+1}$, hence, we have $y^{\mathcal{I},t+1} = e * y^{\mathcal{I},t+1}$, and the lemma follows.

Otherwise, we have $index(e) \in \mathcal{F} - \mathcal{I}$. Then, for any $S \in \mathcal{P}_{t+1}$, Equation (5.7) gives

$$(y^{\mathcal{I},t+1} - e * y^{\mathcal{I},t+1})_S = \begin{cases} 0 & \text{if } C(e) \cap S \neq \emptyset \\ \frac{1}{t+3} & \text{if } C(e) \cap S = \emptyset \end{cases} \quad (5.9)$$

$$= \frac{1}{t+3} \mathbf{1}_S^{E-C(e),t+1} \quad (5.10)$$

The good-decomposition property of G implies that $\mathbf{1}^{E-C(e),t+1}$ is a feasible integral solution of $SA^t(ATSP_{BAL}(G))$. \square

Lemma 5.3.8 *We have $y^{\emptyset,0} \in SA^0(ATSP_{BAL}(G))$.*

Proof. Observe that $y^{\emptyset,0}$ has $|E| + 1$ elements, and $y_{\emptyset}^{\emptyset,0} = 1$ (by Definition 5.3.2); the other $|E|$ elements are indexed by the singleton sets of E . For notational convenience, let $y \in \mathbb{R}^E$ denote the restriction of $y^{\emptyset,0}$ to indices that are singleton sets; thus, $y_e = y_{\{e\}}^{\emptyset,0}$, $\forall e \in E$. By Definition 5.3.2, $y_e = 1/2$ if $e \in E(C_j)$ where $j \in \mathcal{F}$, and $y_e = 1$, otherwise. We claim that y is a feasible solution to $\widehat{ATSP}_{BAL}(G)$.

y is clearly in $[0, 1]^E$. Moreover, y satisfies the balance-constraint at each vertex because it assigns the same value (either $1/2$ or 1) to every edge in a dicycle C_j , $\forall j \in \mathcal{N}$.

To show feasibility of the cut-constraints, consider any cut $\emptyset \neq U \subsetneq V$. Since $\mathbf{1}^E$ is a feasible solution, there exists an edge $e \in E$ crossing from U to $V - U$. If $e \in E(C_j)$, $j \in \mathcal{F}$,

then we have $y_e = 1$, which implies $y(\delta^{out}(U)) = y(\delta^{in}(U)) \geq 1$ (from the balance-constraints at the vertices). Otherwise, we have $e \in E(C_j), j \in \mathcal{F}$. Applying the good-decomposition property of G , we see that there exists an edge $e' (\neq e) \in E - E(C_j)$ such that $e' \in \delta^{out}(U)$, i.e., $|\delta^{out}(U)| \geq 2$. Since $y_e \geq \frac{1}{2}$ for each $e \in E$, the cut-constraints $y(\delta^{in}(U)) = y(\delta^{out}(U)) \geq 1$ are satisfied. \square

The next result presents our first lower bound on the integrality ratio for the level- t relaxation of the Sherali-Adams system starting with the balanced LP relaxation. The relevant instance is a simple digraph on $\Theta(t)$ vertices; see Figure 5.1. In the next subsection, we present better integrality ratios using the CGK construction, but the CGK digraph is not as simple and it has $\Theta(t^t)$ vertices.

Theorem 5.3.9 *Let t be a nonnegative integer, and let $\epsilon \in \mathbb{R}$ satisfy $0 < \epsilon \ll 1$. There exists a digraph on $\nu = \nu(t, \epsilon) = \Theta(t/\epsilon)$ vertices such that the integrality ratio for the level- t tightening of the balanced LP relaxation (Bal LP) (by the Sherali-Adams system) is $\geq 1 + \frac{1-\epsilon}{2t+3}$.*

Proof. Let G be the digraph together with the good decomposition shown in Figure 5.1, and let the cost of each edge in G be 1. We call an edge of G a *thin* edge if it is contained in a dicycle of length 2; we call the other edges of G the *thick* edges; see the illustration in Figure 5.1. Consider the metric completion H of G . It can be seen that the optimal value of an integral solution of ATSP on H (equivalent to the minimum-cost Eulerian subdigraph of G) is $\geq 4\ell + 2$, where ℓ is the length of the “middle path.” (This can be proved by induction on ℓ , using similar arguments as in Cheung [18, Claim 3 of Theorem 11].)

Given t and ϵ , we fix $\ell = 2(2t + 3)/\epsilon$ to get a digraph G (and its edge costs) from the above family.

By Corollary 5.3.4 the fractional solution $y^{\emptyset, t}$ (Definition 5.3.2) is in $\text{SA}^t(\widehat{\text{ATSP}}_{\text{BAL}}(G))$: we have $y_e^{\emptyset, t} = 1$ for each thick edge e , and $y_e^{\emptyset, t} = \frac{t+1}{t+2}$ for each thin edge e . By Section 5.2.2, we can extend $y^{\emptyset, t}$ to a feasible solution of $\text{SA}^t(\widehat{\text{ATSP}}_{\text{BAL}}(H))$.

Hence, the integrality ratio is

$$\geq \frac{4\ell + 2}{2\ell + 4 + 2\ell \frac{t+1}{t+2}} \geq \frac{2(t+2)}{2t+3} - \frac{2}{\ell} \geq 1 + \frac{1-\epsilon}{2t+3}.$$

\square

5.3.2 CGK (Charikar-Goemans-Karloff) construction

We briefly explain the CGK [10] construction and show in Theorem 5.3.14 that the resulting digraph has a good decomposition. This theorem along with a lemma from [10] shows that the integrality ratio is $\geq 1 + \frac{1-\epsilon}{t+1}$ for level t of the Sherali-Adams system starting with the balanced LP relaxation, for any given $0 < \epsilon \ll 1$, see Theorem 5.3.16.

Let r be a fixed positive integer. Let G_0 be the digraph with a single vertex. Let G_1 consist of a bidirected path of $r + 2$ vertices, starting at the “source” p and ending at the “sink” q , whose $2(r + 1)$ edges have cost 1 (see Figure 5.2). We call $E(G_1)$ the *external edge set* of G_1 (we use this in the proof of Lemma 5.3.13).

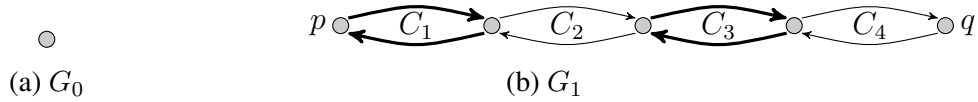


Figure 5.2: G_0 and G_1 for $r = 3$

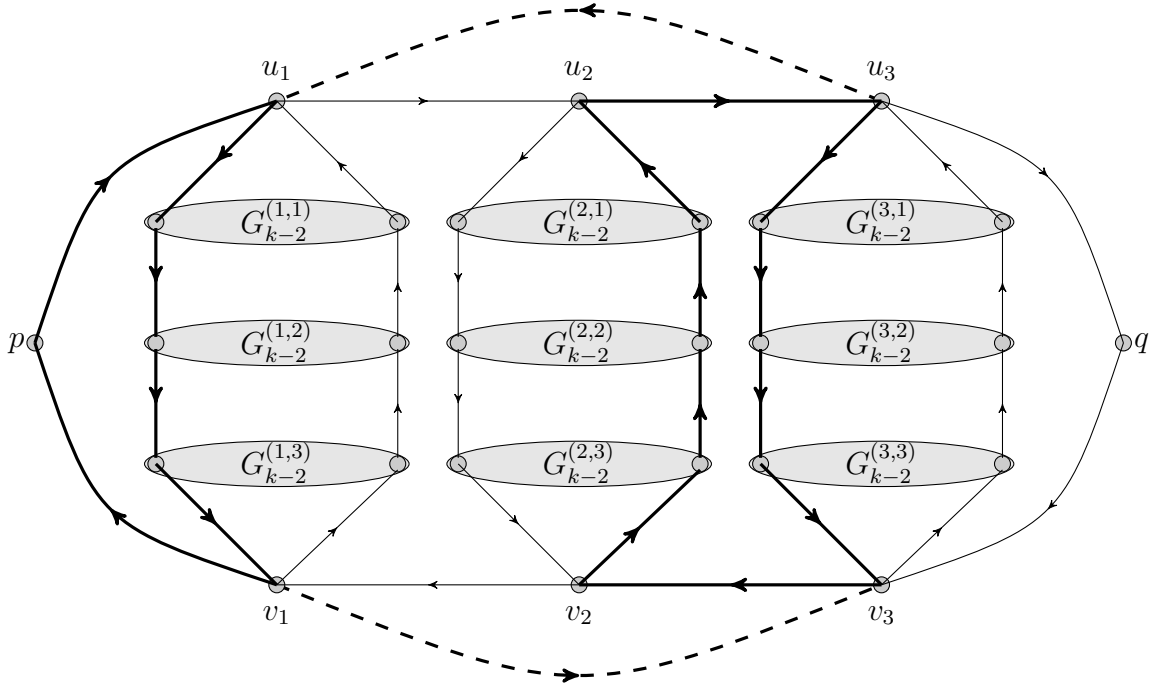


Figure 5.3: G_k and L_k for $k \geq 2$ and $r = 3$

For each $k \geq 2$, we construct G_k by taking r copies of G_{k-1} , additional source and sink vertices p and q , a dipath from p to q of $r + 1$ edges visiting the sources of the r copies in the order u_1, u_2, \dots, u_r , and another dipath from q to p of $r + 1$ edges visiting the sinks of the r copies in the order v_r, v_{r-1}, \dots, v_1 where u_i, v_i denote the source and sink of the i th copy of G_{k-1} (see Figure 5.3). All the new edges have cost r^{k-1} . Denote the i -th copy of G_{k-1} by $G_{k-1}^{(i)}$. Let $E_k = E(G_k) - \bigcup_{1 \leq i \leq r} E(G_{k-1}^{(i)})$. Let $\{G_{k-2}^{(i,j)}\}_{1 \leq j \leq r}$ be the r copies of G_{k-2} in $G_{k-1}^{(i)}$. Let $E_{k-1}^{(i)} = E(G_{k-1}^{(i)}) - \bigcup_{1 \leq j \leq r} E(G_{k-2}^{(i,j)})$. Let $A^{(i)}$ be the dipath from u_i to v_i in $E_{k-1}^{(i)}$ and let $B^{(i)}$ be the dipath from v_i to u_i in $E_{k-1}^{(i)}$. Let $E_{k-1}^{[r]} = \bigcup_{1 \leq i \leq r} E_{k-1}^{(i)}$. We call $E_k \cup E_{k-1}^{[r]}$ the *external* edge set of G_k . The other edges form the *internal* edge set of G_k .

For each $k \geq 2$, the digraph L_k is constructed from G_k by removing vertices p and q , and adding the edges (u_r, u_1) and (v_1, v_r) , both of cost r^{k-1} . Let $E'_k = (E_k \cup \{(u_r, u_1), (v_1, v_r)\}) - \{(p, u_1), (v_1, p), (u_r, q), (q, v_r)\}$. We call $E'_k \cup E_{k-1}^{[r]}$ the *external* edge set of L_k . The other edges form the *internal* edge set of L_k . (Our description of the CGK construction is essentially the same as in [10], but they use s and t to denote the source and sink vertices, whereas we use p and q ; this is to avoid conflict with our symbol t for the number of levels of the SA system.)

Fact 5.3.10 *Let $k \geq 2$ be a positive integer. The external edge set of L_k , i.e., $E'_k \cup E_{k-1}^{[r]}$, can be partitioned into r dicycles C'_1, \dots, C'_r such that*

$$\begin{aligned} C'_i &= \{(u_i, u_{i+1}), (v_{i+1}, v_i)\} \cup B^{(i)} \cup A^{(i+1)}, \text{ for } 1 \leq i \leq r-1, \text{ and} \\ C'_r &= \{(u_r, u_1), (v_1, v_r)\} \cup B^{(r)} \cup A^{(1)}. \end{aligned}$$

Moreover, for each dicycle C'_i , $i = 1, \dots, r$, $L_k - E(C'_i)$ is strongly connected.

We denote the decomposition of the external edge set of L_k by

$$\mathcal{C}_{L_k}(E'_k \cup E_{k-1}^{[r]}) = \{C'_1, \dots, C'_r\}.$$

Fact 5.3.11 *Let $k \geq 2$ be a positive integer. The external edge set of G_k , i.e., $E_k \cup E_{k-1}^{[r]}$, can be partitioned into $r + 1$ dicycles C_0, C_1, \dots, C_r such that*

$$\begin{aligned} C_i &= \{(u_i, u_{i+1}), (v_{i+1}, v_i)\} \cup B^{(i)} \cup A^{(i+1)}, \text{ for } 1 \leq i \leq r-1, \\ C_0 &= \{(p, u_1), (v_1, p)\} \cup A^{(1)}, \text{ and} \end{aligned}$$

$$C_r = \{(u_r, q), (q, v_r)\} \cup B^{(r)}.$$

Moreover, for each dicycle C_i , $i = 0, 1, \dots, r$, $G_k - E(C_i)$ has two strongly-connected components, where one contains the source p and the other one contains the sink q .

We denote the decomposition of the external edge set of G_k by

$$\mathcal{C}_{G_k}(E_k \cup E_{k-1}^{[r]}) = \{C_0, C_1, \dots, C_r\}.$$

Next we identify a structural property that will allow us to prove that L_k has a good decomposition.

Definition 5.3.12 We say that G_k has a p, q good decomposition, if the edge set of G_k can be partitioned into dicycles C_1, C_2, \dots, C_N such that for each $1 \leq i \leq N$, either

- (1) C_i consists of external edges, and moreover, $G_k - E(C_i)$ has two strongly connected components, one containing the source p and the other one containing the sink q .
- (2) C_i consists of internal edges of G_k , and moreover, $G_k - E(C_i)$ is strongly connected.

Lemma 5.3.13 For all $k \geq 1$, G_k has a p, q good decomposition.

Proof. We prove the result by strong induction on k . For the base cases, consider G_1 and G_2 . For G_1 , we take the dicycles C_1, \dots, C_N to be the length 2 dicycles formed by two anti-parallel edges; thus, $N = r + 1$ (see Figure 5.2). For G_2 , we use the decomposition of the external edge set given by Fact 5.3.11.

For the induction step, we have $k \geq 3$; we assume that the statement holds for $1, 2, \dots, k - 1$ and prove that it holds for k . By the induction hypothesis, for each $1 \leq i, j \leq r$, we know that $G_{k-2}^{(i,j)}$ has a p, q good decomposition $\mathcal{C}(E(G_{k-2}^{(i,j)})) = \{C_1^{(i,j)}, C_2^{(i,j)}, \dots, C_{N(i,j)}^{(i,j)}\}$. Consider the decomposition of $E(G_k)$ into edge-disjoint dicycles given by $\widehat{\mathcal{C}} = \mathcal{C}_{G_k}(E_k \cup E_{k-1}^{[r]}) \cup \bigcup_{1 \leq i, j \leq r} \mathcal{C}(E(G_{k-2}^{(i,j)}))$. We claim that $\widehat{\mathcal{C}}$ is a p, q good decomposition of G_k . Clearly, for $C \in \widehat{\mathcal{C}}$ such that $E(C) \subseteq E_k \cup E_{k-1}^{[r]}$, we are done by Fact 5.3.11. Now, consider one of the other dicycles $C \in \widehat{\mathcal{C}}$; thus C consists of some internal edges of G_k . Then, there exists an i and j ($1 \leq i, j \leq r$) such that $C \in \mathcal{C}(E(G_{k-2}^{(i,j)}))$. We have two cases, since either condition (1) or (2) of p, q good decomposition of $G_{k-2}^{(i,j)}$ applies to C . In the first case, $G_{k-2}^{(i,j)} - E(C)$ has two strongly

connected components, where one contains the source $p^{(i,j)}$ of $G_{k-2}^{(i,j)}$ and the other one contains the sink $q^{(i,j)}$ of $G_{k-2}^{(i,j)}$. Note that the external edge set of G_k “strongly connects” $p^{(i,j)}$ and $q^{(i,j)}$, hence, $G_k - E(C)$ is strongly connected. In the second case, $G_{k-2}^{(i,j)} - E(C)$ is strongly connected; then clearly, $G_k - E(C)$ is strongly connected. Thus \hat{C} is a p, q good decomposition of G_k . \square

Theorem 5.3.14 *For $k \geq 2$, L_k has a good decomposition with witness set \mathcal{F} such that $\mathcal{F} = \mathcal{N}$, i.e. every edge in any cycle in the decomposition can be assigned a fractional value.*

Proof. Let $\mathcal{C}_{L_k}(E'_k \cup E_{k-1}^{[r]})$ be the decomposition of the external edge set of L_k given by Fact 5.3.10. If $k = 2$, then we are done (we have a good decomposition of L_k with $\mathcal{F} = \mathcal{N}$). Otherwise, we use the decomposition $\hat{C} = \mathcal{C}_{L_k}(E'_k \cup E_{k-1}^{[r]}) \cup \bigcup_{1 \leq i, j \leq r} \mathcal{C}(E(G_{k-2}^{(i,j)}))$, where $\mathcal{C}(E(G_{k-2}^{(i,j)}))$ is a p, q good decomposition of $G_{k-2}^{(i,j)}$. Using similar arguments as in the proof of Lemma 5.3.13, it can be seen that \hat{C} is a good decomposition with $\mathcal{F} = \mathcal{N}$. \square

Lemma 5.3.15 (Lemma 3.2 [10]) *For $k \geq 2$ and $r \geq 3$, the minimum cost of the Eulerian subdigraph of L_k is $\geq (2k - 1)(r - 1)r^{k-1}$.*

Theorem 5.3.16 *Let t be a nonnegative integer, and let $\epsilon \in \mathbb{R}$ satisfy $0 < \epsilon \ll 1$. There exists a digraph on $\nu = \nu(t, \epsilon) = O((t/\epsilon)^{(t/\epsilon)})$ vertices such that the integrality ratio for the level- t tightening of the balanced LP relaxation for ATSP(Bal LP) (by the Sherali-Adams system) is $\geq 1 + \frac{1-\epsilon}{t+1}$.*

Proof. Given t and ϵ , we apply the CGK construction with $k = r = 5(t+1)/\epsilon$ to get the digraph L_k and its edge costs. Let H_k be the metric completion of L_k .

We know from CGK [10] that the total cost of the edges in L_k is $\leq 2k(r+1)r^{k-1}$. By Theorem 5.3.14, L_k has a good decomposition C_1, \dots, C_N such that each of the dicycles C_j has its index in the witness set \mathcal{F} (informally, each edge is assigned to a fractional dicycle). Hence, Corollary 5.3.4 implies that the fractional solution that assigns the value $\frac{t+1}{t+2}$ to (the variable of) each edge is feasible for $\text{SA}^t(\widehat{\text{ATSP}}_{\text{BAL}}(L_k))$. By Section 5.2.2, this feasible solution can be extended to a feasible solution in $\text{SA}^t(\widehat{\text{ATSP}}_{\text{BAL}}(H_k))$.

Then, using Lemma 5.3.15, we see that the integrality ratio of $\text{SA}^t(\widehat{\text{ATSP}}_{\text{BAL}}(H_k))$ is

$$\begin{aligned} &\geq \frac{(2k-1)(r-1)r^{k-1}}{\left(\frac{t+1}{t+2}\right)2k(r+1)r^{k-1}} = 1 + \frac{1}{t+1} - \frac{5r-1}{\frac{t+1}{t+2}(r+1)(2r)} \geq 1 + \frac{1}{t+1} - \frac{5}{\frac{t+1}{t+2}} \frac{1}{(2r)} \\ &\geq 1 + \frac{1-\epsilon}{t+1}. \end{aligned}$$

\square

5.4 SA applied to the standard LP (DFJ LP) relaxation of ATSP

Let $G = (V, E)$ be a strongly connected digraph that has a good decomposition, and moreover, has both indegree and outdegree ≤ 2 for every vertex. We use the same notation as in Section 5.3.1, i.e., C_1, C_2, \dots, C_N denote the edge disjoint dicycles of the decomposition, and there exists $\mathcal{F} \subseteq \mathcal{N} = \{1, \dots, N\}$ such that \mathcal{F} is nonempty and $G - E(C_j)$ is strongly connected for all $j \in \mathcal{F}$.

We define a vertex-splitting operation that splits every vertex that has indegree 2 (and outdegree 2) into two vertices (along with some edges); our definition depends on the given good decomposition of the digraph. The purpose of the vertex-splitting operation will be clear from Fact 5.4.1.

Vertex-splitting Operation: Let $v \in V(G)$ whose indegree and outdegree is 2. Suppose C_i, C_j are the dicycles in the good decomposition going through v . Let $e_{i1} = (v_{i1}, v), e_{j1} = (v_{j1}, v)$ and $e_{i2} = (v, v_{i2}), e_{j2} = (v, v_{j2})$ be the edges in $\delta^{in}(v), \delta^{out}(v)$, respectively, where $e_{i1}, e_{i2} \in C_i$ and $e_{j1}, e_{j2} \in C_j$. We split v into v^u, v^b as follows:

- Replace e_{i1}, e_{j2} by $e_{i1}^{new} = (v_{i1}, v^u), e_{j2}^{new} = (v^u, v_{j2})$ (the new edges are called solid edges)
- Replace e_{i2}, e_{j1} by $e_{i2}^{new} = (v^b, v_{i2}), e_{j1}^{new} = (v_{j1}, v^b)$ (the new edges are called solid edges)
- Add the auxiliary edges (also called dashed edges) $e_0 = (v^b, v^u), e'_0 = (v^u, v^b)$.

See Figure 5.4 for an illustration.

We obtain $G^{new} = (V^{new}, E^{new})$ from G by applying the vertex-splitting operation to every vertex in G whose indegree and outdegree is 2. We map each dicycle $C_j, j \in \mathcal{N}$, of G to a set of edges of G^{new} that we call a cycle and that we will (temporarily) denote by C_j^{new} . We define C_j^{new} to be the following set of edges: for every edge of C_j , its image (in G^{new}) is in C_j^{new} ; moreover, for every splitted vertex v of G incident to C_j , note that one of v^u or v^b (the two images of v) is the head of one of the two edges of C_j^{new} incident to $\{v^u, v^b\}$, and one of the two auxiliary edges e_0, e'_0 has its head at the same vertex; we place this auxiliary edge also in C_j^{new} . For example, in Figure 5.4, the cycle C_i^{new} contains the edges e_{i1}^{new} (image of e_{i1}), e_{i2}^{new} (image of e_{i2}), and the auxiliary edge e_0 , whereas the cycle C_j^{new} contains the edges $e_{j1}^{new}, e_{j2}^{new}$, and the auxiliary edge e'_0 .

In what follows, we simplify the notation for the cycles of G^{new} to C_j (rather than C_j^{new}); there is some danger of ambiguity, but the context will resolve this. We denote the set of auxiliary

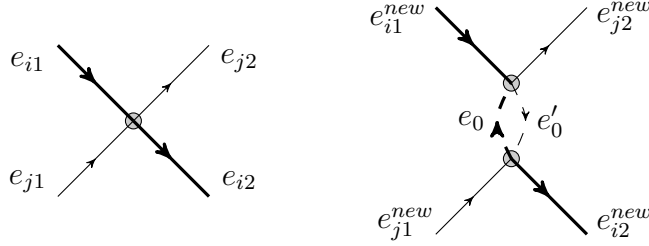


Figure 5.4: An illustration of the vertex-splitting operation used for mapping G to G^{new} .

edges (also called the dashed edges) of a cycle $C_j = C_j^{new}$ by $D(C_j)$, and we denote the set of remaining edges of $C_j = C_j^{new}$ by $E(C_j)$. Note that $E^{new} = E(G^{new}) = \bigcup_{j \in \mathcal{N}} (E(C_j) \cup D(C_j))$. Clearly, there is a bijection between the edges of $E(C_j) = E(C_j^{new})$ in G^{new} and the edges of $E(C_j)$ in G . Also, observe that in G^{new} , the dashed edges are partitioned among the cycles $C_j^{new}, j \in \mathcal{N}$.

Fact 5.4.1 Consider a digraph $G = (V, E)$ that has a good decomposition, and consider $x \in \mathbb{R}^E$ such that (1) $0 \leq x \leq 1$, (2) for every dicycle $C_j, j \in \mathcal{N}$, x_e is the same for all edges e of C_j , and (3) for every vertex v with indegree = 1 = outdegree, $x(\delta^{in}(v)) = x(\delta^{out}(v)) = 1$. Then, for the digraph $G^{new} = (V^{new}, E^{new})$ obtained by applying the vertex-splitting operations, there exists $x^{new} \in \mathbb{R}^{E^{new}}$ such that $0 \leq x^{new} \leq 1$, and $x^{new}(\delta^{in}(v)) = x^{new}(\delta^{out}(v)) = 1, \forall v \in V^{new}$.

Proof. For each $j \in \mathcal{N}$, we consider the dicycle C_j . Let α_j be the x -value associated with the dicycle C_j of G , i.e., $x_e = \alpha_j, \forall e \in E(C_j)$. Then, in x^{new} and G^{new} , we fix $x_e = \alpha_j, \forall e \in E(C_j) = E(C_j^{new})$, and we fix $x_e = (1 - \alpha_j), \forall e \in D(C_j) = D(C_j^{new})$. It can be seen that x^{new} satisfies the given conditions. \square

Definition 5.4.2 Consider the digraph G^{new} . For any $j \in \mathcal{F}$, let

$$tour(j) := D(C_j) \cup \bigcup_{i \in (\mathcal{N} - j)} E(C_i).$$

Thus $tour(j)$ consists of all the solid edges except those in C_j together with all the dashed edges of C_j . Note that each vertex in G^{new} has exactly one incoming edge and exactly one outgoing edge in $tour(j)$. Thus $tour(j)$ forms a set of vertex-disjoint dicycles that partition V^{new} .

Definition 5.4.3 Let G be a digraph with indegree and outdegree ≤ 2 at every vertex, and suppose that G has a good decomposition with witness set \mathcal{F} . Let G^{new} be the digraph obtained by applying vertex-splitting operations to G and its good decomposition. Then G is said to have the *good tours property* if $\text{tour}(j)$ is connected (i.e., $\text{tour}(j)$ forms a Hamiltonian dicycle of G^{new}) for each $j \in \mathcal{F}$.

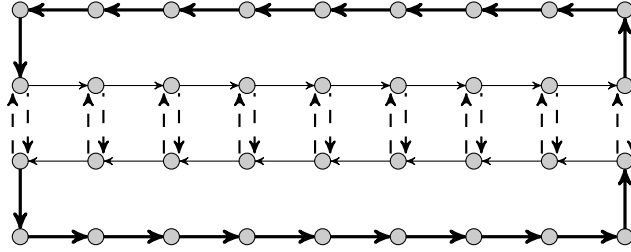
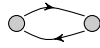
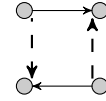


Figure 5.5: Digraph from Figure 5.1 after the vertex-splitting operation



(a)



(b)

Figure 5.6: Transforming a dicycle C_j formed by an anti-parallel pair of thin edges in Figure 5.1 to C_j^{new} by the vertex-splitting operation.

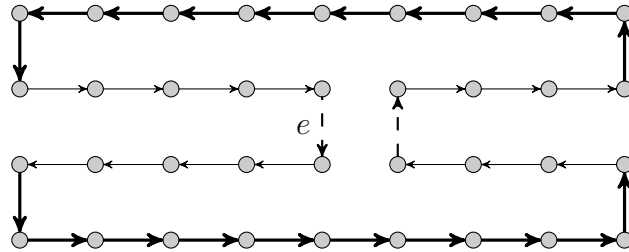


Figure 5.7: $\text{tour}(e)$

5.4.1 Certifying a feasible solution

In what follows, we assume that G is a digraph that satisfies the conditions stated in Definition 5.4.3. We focus on the digraph G^{new} obtained by applying vertex-splitting operations to G ; observe that G^{new} depends on G as well as on the given good decomposition of G . Let $\widehat{\text{ATSP}}_{DFJ}(G^{new})$ be the associated cone of $\widehat{\text{ATSP}}_{DFJ}(G^{new})$.

Let E denote the set of images of the edges of G (the solid edges), and let D denote the set of auxiliary edges (the dashed edges). Given $S \subseteq E$ and $\mathcal{I} \subseteq \mathcal{F}$, let $F^{\mathcal{I}}(S)$ denote the set of indices $j \in \mathcal{F} - \mathcal{I}$ such that $E(C_j)$ intersects S , and let $f^{\mathcal{I}}(S)$ denote the size of this set; thus, $f^{\mathcal{I}}(S)$ denotes the number of “fractional cycles” that intersect S in the solid edges.

Note that each (solid or dashed) edge e is in a unique cycle $C(e)$; let $\text{index}(e)$ denote the index of $C(e)$ in \mathcal{N} ; if $\text{index}(e) \in \mathcal{F} - \mathcal{I}$, then we may use $\text{tour}(e)$ to denote $\text{tour}(\text{index}(e))$.

Let t be a nonnegative integer. We define the feasible solution y for the level- t tightening of the DFJ LP relaxation (of ATSP, by the SA system) as follows:

Definition 5.4.4 For a nonnegative integer t and for any subset \mathcal{I} of \mathcal{F} , let $y^{\mathcal{I},t}$ be a vector indexed by the elements of \mathcal{P}_{t+1} and defined as follows:

$$(y^{\mathcal{I},t})_S = \begin{cases} \frac{t+2-f^{\mathcal{I}}(S)}{t+2} & \text{if } S \cap D = \emptyset \quad (S \text{ has no dashed edges}) \\ \frac{1}{t+2} & \text{if } S \cap D \neq \emptyset \text{ and } \exists i \in \mathcal{F} - \mathcal{I} : \text{tour}(i) \supseteq S \\ & (S \text{ contains some dashed edges and is contained in a tour}) \\ 0 & \text{otherwise} \end{cases} \quad (5.11)$$

Observe that the second case applies when the set S has one or more dashed edges, and moreover, S is contained in a $\text{tour}(i)$, $i \in \mathcal{F} - \mathcal{I}$; also, observe that there is at most one tour that contains S , because the dashed edges are partitioned among the cycles C_j , $j \in \mathcal{N}$, so each dashed edge in S belongs to a unique tour.

Theorem 5.4.5 Let $G = (V, E)$ be a strongly connected digraph that has a good decomposition with witness set \mathcal{F} , and moreover, has (i) both indegree and outdegree ≤ 2 for every vertex, and (ii) satisfies the “good tours” property. Then, for any nonnegative integer t , and any $\mathcal{I} \subseteq \mathcal{F}$ with $|\mathcal{I}| \leq |\mathcal{F}| - (t + 2)$, we have

$$y^{\mathcal{I},t} \in \text{SA}^t(\widehat{\text{ATSP}}_{DFJ}(G^{new})).$$

Proof. By Definition 5.4.4, $y_{\emptyset}^{\mathcal{I},t} = 1$. Thus, we only need to prove $y^{\mathcal{I},t} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{new}))$. The proof is by induction on t . The base case is important, and it follows easily from the good decomposition property and the “good tours” property of G . This is done in Lemma 5.4.6 below, where we show that $y^{\mathcal{I},0} \in \text{SA}^0(\text{ATSP}_{DFJ}(G^{new}))$, $\forall \mathcal{I} \subseteq \mathcal{F}, |\mathcal{I}| \leq |\mathcal{F}| - 2$.

In the induction step, we assume that $y^{\mathcal{I},t} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{new}))$ for some integer $t \geq 0$ (the induction hypothesis), and we apply the recursive definition based on the shift operator, namely, $y^{\mathcal{I},t+1} \in \text{SA}^{t+1}(\text{ATSP}_{DFJ}(G^{new}))$ iff for each $e \in E^{new}$

$$e * y^{\mathcal{I},t+1} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{new})), \quad (5.12)$$

$$y^{\mathcal{I},t+1} - e * y^{\mathcal{I},t+1} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{new})). \quad (5.13)$$

Lemma 5.4.8 (below) proves (5.12) and Lemma 5.4.10 (below) proves (5.13). \square

The next lemma proves the base case for the induction; it follows from the “good tours” property of the digraph.

Lemma 5.4.6

$$y^{\mathcal{I},0} \in \text{SA}^0(\text{ATSP}_{DFJ}(G^{new})), \quad \forall \mathcal{I} \subseteq \mathcal{F}, |\mathcal{I}| \leq |\mathcal{F}| - 2.$$

Proof. Note that $y_{\emptyset}^{\mathcal{I},0} = 1$. Let z be the subvector of $y^{\mathcal{I},0}$ on the singleton sets $\{e_i\}$. We need to prove that z is a feasible solution of the DFJ LP relaxation. It can be seen that z is as follows: if $\text{index}(e) \in \mathcal{F} - \mathcal{I}$, then $z_e = \frac{1}{2}$, otherwise, if $e \in E$ (e is a solid edge), then $z_e = 1$, otherwise, if $e \in D$ (e is a dashed edge), then $z_e = 0$. Clearly, z is in $[0, 1]^{E^{new}}$ and satisfies the degree constraints. Now, we need to verify that z satisfies the cut constraints in the digraph G^{new} . Consider any nonempty set of vertices $U \neq V$, and the cut $\delta^{out}(U)$.

Observe that $|\mathcal{F} - \mathcal{I}| \geq 2$, hence, there are at least two indices i, j such that $i, j \in \mathcal{F} - \mathcal{I}$. Hence, both $\text{tour}(i)$ and $\text{tour}(j)$ exist; moreover, every edge e (either solid or dashed) in either $\text{tour}(i)$ or $\text{tour}(j)$ has $z_e \geq \frac{1}{2}$. Clearly, each of $\text{tour}(i)$ and $\text{tour}(j)$ has at least one edge in $\delta^{out}(U)$. Let e_j be an edge of $\text{tour}(j)$ that is in $\delta^{out}(U)$. If $z_{e_j} = 1$, then we are done, since we have $z(\delta^{out}(U)) \geq z_{e_j} = 1$. Thus, we may assume $z_{e_j} = \frac{1}{2}$. Now, we have two cases.

First, suppose that e_j is a dashed edge. Then, note that the edge of $\text{tour}(i)$ in $\delta^{out}(U)$, call it e_i , is distinct from e_j (since the tours are disjoint on the dashed edges), and again we are done, since $z(\delta^{out}(U)) \geq z_{e_i} + z_{e_j} \geq 1$.

In the remaining case, $e_j \in \text{tour}(j)$ is a solid edge and $z_{e_j} = \frac{1}{2}$. Then, $\text{index}(e_j) \in \mathcal{F} - \mathcal{I}$, and so $\text{tour}(e_j)$ exists and it has at least one edge e' in $\delta^{out}(U)$; moreover, $e' \neq e_j$ because $\text{tour}(e_j)$

contains none of the solid edges of the cycle $C_{\text{index}(e_j)}$. Thus, we are done, since $z(\delta^{\text{out}}(U)) \geq z_{e_j} + z_{e'} \geq 1$. It follows that z satisfies all of the cut constraints. \square

The following fact summarizes some easy observations; this fact is used in the next lemma.

Fact 5.4.7 *Let \mathcal{I} be a subset of \mathcal{F} . Suppose that S is not contained in any tour(j), $j \in \mathcal{F} - \mathcal{I}$. (1) Then, for any edge e , $S + e$ is also not contained in any tour(j), $j \in \mathcal{F} - \mathcal{I}$. (2) Similarly, for any index $h \in \mathcal{F}$, S is not contained in any tour(j), $j \in \mathcal{F} - (\mathcal{I} + h)$.*

Lemma 5.4.8 *Suppose that for any nonnegative integer t and any $\mathcal{I}' \subseteq \mathcal{F}$ with $|\mathcal{I}'| \leq |\mathcal{F}| - (t + 2)$, we have $y^{\mathcal{I}', t} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{\text{new}}))$. Then for any $\mathcal{I} \subseteq \mathcal{F}$ with $|\mathcal{I}| \leq |\mathcal{F}| - (t + 3)$,*

$$e * y^{\mathcal{I}, t+1} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{\text{new}})), \quad \forall e \in E^{\text{new}}.$$

Proof. For any edge e and any $S \in \mathcal{P}_{t+1}$, the definition of the shift operator gives

$$(e * y^{\mathcal{I}, t+1})_S = y_{S+e}^{\mathcal{I}, t+1}.$$

Let $C(e)$ denote the cycle containing edge e , and let $\text{index}(e)$ denote the index of $C(e)$ in \mathcal{N} .

We will show that

$$(e * y^{\mathcal{I}, t+1})_S = \begin{cases} y_S^{\mathcal{I}, t+1} & \text{if } e \in E(C_j) \text{ where } j \in \mathcal{I} \cup \overline{\mathcal{F}} \\ & (e \text{ is a solid, integral edge}) \\ 0 & \text{if } e \in D(C_j) \text{ where } j \in \mathcal{I} \cup \overline{\mathcal{F}} \\ & (e \text{ is a dashed, integral edge}) \\ \frac{t+2}{t+3} y_S^{\mathcal{I} + \text{index}(e), t} & \text{if } e \in E(C_j) \text{ where } j \in \mathcal{F} - \mathcal{I} \\ & (e \text{ is a solid, fractional edge}) \\ \frac{1}{t+3} \mathbf{1}_S^{\text{tour}(e), t+1} & \text{if } e \in D(C_j) \text{ where } j \in \mathcal{F} - \mathcal{I} \\ & (e \text{ is a dashed, fractional edge}) \end{cases} \quad (5.14)$$

Lemma 5.4.9 (below) shows that

$$y_S^{\mathcal{I}, t+1} = \frac{t+2}{t+3} y_S^{\mathcal{I} + h, t} + \frac{1}{t+3} \mathbf{1}_S^{\text{tour}(h), t+1}, \quad \forall h \in \mathcal{F} - \mathcal{I}.$$

Hence, for every edge e (i.e., in every case), $e * y^{\mathcal{I}, t+1}$ is in $\text{SA}^t(\text{ATSP}_{DFJ}(G^{\text{new}}))$.

Case 1. $e \in E(C_j)$ where $j \in \mathcal{I} \cup \overline{\mathcal{F}}$ (e is a solid, integral edge). We apply Definition 5.4.4 (the definition of y), and consider the three cases in it:

Subcase 1.1. $S \cap D = \emptyset$. Then we have $(S + e) \cap D = \emptyset$, and moreover, we have $f^{\mathcal{I}}(S) = f^{\mathcal{I}}(S + e)$ (the number of “fractional cycles” intersecting $S \cap E$ and $(S + e) \cap E$ is the same, since e is a non-fractional edge). Hence, $y_{(S+e)}^{\mathcal{I}, t+1} = y_S^{\mathcal{I}, t+1}$.

Subcase 1.2. $S \cap D \neq \emptyset$ and $\exists i \in \mathcal{F} - \mathcal{I} : \text{tour}(i) \supseteq S$. Then it is clear that $(S + e) \cap D \neq \emptyset$ and $\text{tour}(i) \supseteq S + e$, because $\text{tour}(i)$ contains every solid edge except those in the fractional cycle C_i . Hence, $y_{S+e}^{\mathcal{I}, t+1} = \frac{1}{t+3} = y_S^{\mathcal{I}, t+1}$.

Subcase 1.3. $S \cap D \neq \emptyset$ and $\forall j \in \mathcal{F} - \mathcal{I} : \text{tour}(j) \not\supseteq S$. Then it is easily seen that both conditions apply to $S + e$ (rather than S). Hence, $y_{S+e}^{\mathcal{I}, t+1} = 0 = y_S^{\mathcal{I}, t+1}$.

Case 2. We have $e \in D(C_j)$ where $j \in \mathcal{I} \cup \overline{\mathcal{F}}$ (e is a dashed, integral edge). We apply Definition 5.4.4, noting that $(S + e) \cap D \neq \emptyset$ and there exists no index $i \in \mathcal{F} - \mathcal{I}$ such that $\text{tour}(i) \supseteq S + e$ (no “valid tour” contains a dashed, integral edge), hence, $y_{S+e}^{\mathcal{I}, t+1} = 0$.

Case 3. We have $e \in E(C_j)$ where $j \in \mathcal{F} - \mathcal{I}$ (e is a solid, fractional edge). We apply Definition 5.4.4. We have two subcases, either $S \cap D = \emptyset$, or not.

Subcase 3.1. If $S \cap D = \emptyset$, then $(S + e) \cap D = \emptyset$. Thus, the analysis is the same as in the previous section; in particular, see Equation (5.6) in the proof of Lemma 5.3.6. Hence, we have $y_{S+e}^{\mathcal{I}, t+1} = \frac{t+2}{t+3} y_S^{\mathcal{I} + \text{index}(e), t}$.

Subcase 3.2. Otherwise, $S \cap D \neq \emptyset$. Then we have two further subcases: either there is an $i \in \mathcal{F} - \mathcal{I}$ with $\text{tour}(i) \supseteq S$ or not.

Subcase 3.2.1 Consider the first subcase; thus, $S \subseteq \text{tour}(i)$ where $i \in \mathcal{F} - \mathcal{I}$. Note that S is not contained in other tours since $S \cap D \neq \emptyset$. We have two further subcases, either $e \in E(C_i)$ or not.

Subcase 3.2.1.1. If $e \in E(C_i)$, then $\text{tour}(i) \not\supseteq (S + e)$, hence, $y_{S+e}^{\mathcal{I}, t+1} = 0$ (by the last case in the definition of y); moreover, note that $\text{tour}(i)$ is the unique tour containing S but it is *not* a “valid tour” w.r.t. $\mathcal{I} + \text{index}(e)$, hence, $y_S^{\mathcal{I} + \text{index}(e), t} = 0$ (by the last case in Definition 5.4.4).

Subcase 3.2.1.2. Otherwise, if $e \notin E(C_i)$, then $\text{tour}(i) \supseteq (S + e)$, and moreover, $\text{tour}(i)$ is a “valid tour” w.r.t. $\mathcal{I} + \text{index}(e)$ (since $i \notin \mathcal{I}$ and $i \neq \text{index}(e)$), hence, we have $y_{S+e}^{\mathcal{I}, t+1} = \frac{1}{t+3} = \frac{t+2}{t+3} \frac{1}{t+2} = \frac{t+2}{t+3} y_S^{\mathcal{I} + \text{index}(e), t}$ (by the second case in Definition 5.4.4, for both LHS and RHS).

Subcase 3.2.2. Consider the last subcase; thus, $S \not\subseteq \text{tour}(i)$ for all $i \in \mathcal{F} - \mathcal{I}$. Then by Fact 5.4.7, the same assertion holds w.r.t. $(S + e)$ (rather than S), as well as w.r.t. $(\mathcal{I} + \text{index}(e))$ (rather than \mathcal{I}). Hence, we have $y_{S+e}^{\mathcal{I}, t+1} = 0 = \frac{t+2}{t+3} y_S^{\mathcal{I} + \text{index}(e), t}$ (by the last case in Definition 5.4.4, for both LHS and RHS).

Case 4. We have $e \in D(C_j)$ where $j \in \mathcal{F} - \mathcal{I}$ (e is a dashed, fractional edge). We apply Definition 5.4.4, noting that $(S + e) \cap D \neq \emptyset$. We have two subcases, either $\text{tour}(e) \supseteq S$, or not. If $\text{tour}(e) \supseteq S$, then the second case of Definition 5.4.4 together with the fourth case of Equation (5.14) (the definition of $e * y$) gives $y_{S+e}^{\mathcal{I}, t+1} = \frac{1}{t+3} = \frac{1}{t+3} \mathbf{1}_S^{\text{tour}(e), t+1}$. Otherwise, $\text{tour}(e) \not\supseteq S$, and then we have $y_{S+e}^{\mathcal{I}, t+1} = 0 = \frac{1}{t+3} \mathbf{1}_S^{\text{tour}(e), t+1}$; note that the last case of Definition 5.4.4 applies because $\text{tour}(e)$ is the unique “valid tour” that could contain e .

□

Lemma 5.4.9 shows that $y^{\mathcal{I}, t+1}$, restricted to \mathcal{P}_{t+1} , is in $\text{SA}^t(\text{ATSP}_{DFJ}(G^{\text{new}}))$; this is used in Lemma 5.4.8 to show that $e * y^{\mathcal{I}, t+1}$ is in $\text{SA}^t(\text{ATSP}_{DFJ}(G^{\text{new}}))$.

Lemma 5.4.9 *For any nonnegative integer t , any $S \in \mathcal{P}_{t+1}$, any $\mathcal{I} \subseteq \mathcal{F}$ with $|\mathcal{I}| \leq |\mathcal{F}| - (t+3)$, and any $h \in \mathcal{F} - \mathcal{I}$, we have*

$$y_S^{\mathcal{I}, t+1} = \frac{t+2}{t+3} y_S^{\mathcal{I}+h, t} + \frac{1}{t+3} \mathbf{1}_S^{\text{tour}(h), t+1}. \quad (5.15)$$

Proof. We have $S \subseteq D \cup E$, $|S| \leq t+1$.

We apply Definition 5.4.4 (the definition of y) to $y^{\mathcal{I}, t+1}$, and we have three cases.

Case 1. $S \cap D = \emptyset$. Then $y_S^{\mathcal{I}, t+1} = \frac{(t+3)-f^{\mathcal{I}}(S)}{t+3}$. For the RHS, we have two subcases, either $\text{tour}(h) \supseteq S$ or not. In the first subcase, we have $S \cap E(C_h) = \emptyset$ (since $\text{tour}(h)$ contains none of the solid edges of C_h), hence, $f^{\mathcal{I}+h}(S) = f^{\mathcal{I}}(S)$, consequently, the RHS is $\frac{t+2}{t+3} \frac{(t+3)-f^{\mathcal{I}}(S)}{t+2} + \frac{1}{t+3}$, which is the same as the LHS. In the other subcase, $\text{tour}(h) \not\supseteq S$. Then, we have $S \cap E(C_h) \neq \emptyset$ (because $S \subseteq E$ and $\text{tour}(h)$ contains all solid edges except those in C_h), hence, $f^{\mathcal{I}+h}(S) = f^{\mathcal{I}}(S) - 1$, and consequently, the RHS is $\frac{t+2}{t+3} \frac{(t+3)-f^{\mathcal{I}}(S)}{t+2} + 0 = \frac{(t+3)-f^{\mathcal{I}}(S)}{t+3}$, which is the same as the LHS.

Case 2. $S \cap D \neq \emptyset$ and there exists $j \in \mathcal{F} - \mathcal{I}$ such that $\text{tour}(j) \supseteq S$. Then $y_S^{\mathcal{I}, t+1} = \frac{1}{t+3}$, by Definition 5.4.4. For the RHS, we have two subcases, either $j = h$ or not. In the first subcase, we have $y_S^{\mathcal{I}+h, t} = 0$, because $\text{tour}(h)$ is the unique tour containing S but it is *not* a “valid tour” w.r.t. $\mathcal{I} + h$, hence, the last case in Definition 5.4.4 applies. Thus, the RHS is $0 + \frac{1}{t+3} \mathbf{1}_S^{\text{tour}(h), t+1} = \frac{1}{t+3}$, which is the same as the LHS. In the second subcase, $j \neq h$. Then, in the RHS, $y_S^{\mathcal{I}+h, t} = \frac{1}{t+2}$, because $j \in \mathcal{F} - (\mathcal{I} + h)$ and $\text{tour}(j) \supseteq S$ so the second case in Definition 5.4.4 applies. Moreover, $\mathbf{1}_S^{\text{tour}(h), t+1} = 0$, because $j \neq h$, and $\text{tour}(j)$ is the unique tour containing S , so $\text{tour}(h) \not\supseteq S$. Thus, the RHS is $\frac{t+2}{t+3} \frac{1}{t+2} + 0 = \frac{1}{t+3}$, which is the same as the LHS.

Case 3. $S \cap D \neq \emptyset$ and $\text{tour}(j) \not\supseteq S, \forall j \in \mathcal{F} - \mathcal{I}$. Then $y_S^{\mathcal{I}, t+1} = 0$. In the RHS, $y_S^{\mathcal{I}+h, t} = 0$, by the third case in Definition 5.4.4, since the relevant conditions hold (by Fact 5.4.7). Moreover, $\mathbf{1}_S^{\text{tour}(h), t+1} = 0$, because $h \in \mathcal{F} - \mathcal{I}$ and $\text{tour}(h) \not\supseteq S$. Thus, the RHS is 0, which is the same as the LHS.

This completes the proof of the lemma. \square

Lemma 5.4.10 Suppose that for any nonnegative integer t and any $\mathcal{I}' \subseteq \mathcal{F}$ with $|\mathcal{I}'| \leq |\mathcal{F}| - (t+2)$, we have $y^{\mathcal{I}', t} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{\text{new}}))$. Then for any $\mathcal{I} \subseteq \mathcal{F}$ with $|\mathcal{I}| \leq |\mathcal{F}| - (t+3)$,

$$y^{\mathcal{I}, t+1} - e * y^{\mathcal{I}, t+1} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{\text{new}})), \quad \forall e \in E^{\text{new}}.$$

Proof. By Lemma 5.4.8 and Lemma 5.4.9, we have for each $e \in E^{\text{new}} = E \cup D$ and any $S \in \mathcal{P}_{t+1}$,

$$(y^{\mathcal{I}, t+1} - e * y^{\mathcal{I}, t+1})_S = \begin{cases} 0 & \text{if } e \in E(C_j) \text{ where } j \in \mathcal{I} \cup \overline{\mathcal{F}} \\ & (e \text{ is a solid, integral edge}) \\ y_S^{\mathcal{I}, t+1} & \text{if } e \in D(C_j) \text{ where } j \in \mathcal{I} \cup \overline{\mathcal{F}} \\ & (e \text{ is a dashed, integral edge}) \\ \frac{1}{t+3} \mathbf{1}_S^{\text{tour}(e), t+1} & \text{if } e \in E(C_j) \text{ where } j \in \mathcal{F} - \mathcal{I} \\ & (e \text{ is a solid, fractional edge}) \\ \frac{t+2}{t+3} y_S^{\mathcal{I} + \text{index}(e), t} & \text{if } e \in D(C_j) \text{ where } j \in \mathcal{F} - \mathcal{I} \\ & (e \text{ is a dashed, fractional edge}) \end{cases} \quad (5.16)$$

Hence, in every case, $y^{\mathcal{I}, t+1} - e * y^{\mathcal{I}, t+1} \in \text{SA}^t(\text{ATSP}_{DFJ}(G^{\text{new}}))$. \square

Theorem 5.4.11 *Let t be a nonnegative integer, and let $\epsilon \in \mathbb{R}$ satisfy $0 < \epsilon \ll 1$. There exists a digraph on $\nu = \nu(t, \epsilon) = \Theta(t/\epsilon)$ vertices such that the integrality ratio for the level- t tightening of the standard LP (DFJ LP) relaxation (for ATSP, by the *Sherali-Adams* system) is $\geq 1 + \frac{1-\epsilon}{2t+3}$.*

Proof. Given t and ϵ , we fix $\ell = 2(2t+3)/\epsilon$ to get a digraph G shown in Figure 5.1 where ℓ is the length of the “middle path”. Let the cost of each edge in G be 1. Then we construct G^{new} from G . We keep the cost of edges in G to be 1 and fix the cost of new edges to be 0. See Figure 5.5; each solid edge has cost 1 and each dashed edge has cost 0. In the proof of Theorem 5.3.9, we claimed that the minimum cost of an Eulerian subdigraph of G is $\geq 4\ell + 2$. It can be seen that the minimum cost of an Eulerian subdigraph of G^{new} is $\geq 4\ell + 2$. (To see this, take an Eulerian subdigraph of G^{new} , then contract all dashed edges contained in it, to get an Eulerian subdigraph of G of the same cost.) Let H be the metric completion of G^{new} . Then, the optimal value of the integral solution in $\text{SA}^t(\widehat{\text{ATSP}}_{DFJ}(H))$ is $\geq 4\ell + 2$.

Now we invoke Theorem 5.4.5, according to which the fractional solution $y^{\emptyset, t}$ (Definition 5.4.4) is in $\text{SA}^t(\widehat{\text{ATSP}}_{DFJ}(G^{new}))$; see Figure 5.5; we have $y_e^{\emptyset, t} = 1$ for each solid, thick edge e (the solid edges of the outer cycle), $y_e^{\emptyset, t} = \frac{t+1}{t+2}$ for each solid, thin edge e (the solid edges of the middle paths), while the value of the dashed edges do not contribute to the value of the objective. By Section 5.2.2, this feasible solution can be extended to a feasible solution in $\text{SA}^t(\widehat{\text{ATSP}}_{DFJ}(H))$.

Hence, the integrality ratio of $\text{SA}^t(\widehat{\text{ATSP}}_{DFJ}(H))$ is

$$\geq \frac{4\ell + 2}{2\ell + 4 + 2\ell \frac{t+1}{t+2}} \geq \frac{2(t+2)}{2t+3} - \frac{2}{\ell} \geq 1 + \frac{1-\epsilon}{2t+3}.$$

□

5.5 Path ATSP

Let $G = (V, E)$ be a digraph with nonnegative edge costs c , and let p and q be two distinguished vertices. We define $\widehat{\text{PATSP}}_{p,q}(G)$ to be the polytope of the following LP that has a variable x_e for each edge e of G :

$$\begin{aligned}
& \text{minimize: } \sum_{e \in E} c_e x_e \\
& \text{subject to: } x(\delta^{in}(S)) \geq 1 & \forall \emptyset \subsetneq S \subseteq V - \{p\} \\
& x(\delta^{out}(S)) \geq 1 & \forall \emptyset \subsetneq S \subseteq V - \{q\} \\
& x(\delta^{in}(v)) = 1 & \forall v \in V - \{p\} \\
& x(\delta^{out}(v)) = 1 & \forall v \in V - \{q\} \\
& x(\delta^{in}(p)) = 0 \\
& x(\delta^{out}(q)) = 0 \\
& 0 \leq x_e \leq 1 & \forall e \in E
\end{aligned}$$

In particular, when G is a complete digraph with metric costs, the above LP is the *standard relaxation for the p - q path ATSP*, which is to compute a Hamiltonian (or, spanning) dipath from p to q with minimum cost in the complete digraph with metric costs. For $\widehat{\text{PATSP}}_{p,q}(G)$, we denote the associated cone by $\text{PATSP}_{p,q}(G)$.

(In the literature, the notation for the two distinguished vertices is s, t , but we use p, q to avoid conflict with our symbol t for the number of levels of the SA system.)

An (p, q) -Eulerian subdigraph \overline{G} of G is V together with a collection of edges of G with multiplicities such that (i) for any $v \in V - \{p, q\}$, the indegree of v equals its outdegree and (ii) the outdegree of p is larger than its indegree by 1 and the indegree of q is larger than its outdegree by 1 and (iii) \overline{G} is weakly connected (i.e., the underlying undirected graph is connected). The p - q path ATSP on the metric completion H of G is equivalent to finding a minimum-cost (p, q) -Eulerian subdigraph of G .

For any subset V' of V , we use $G(V')$ to denote the subdigraph of G induced by V' . As before, we use \mathcal{P}_t to denote $\mathcal{P}_t(E)$ (for the ground set E). Also, by the *restriction* of y on $E' \subseteq E$ we mean the vector $y|_{E'} \in \mathbb{R}^{\mathcal{P}_{t+1}(E')}$ that is given by $(y|_{E'})_S = y_S$ for all $S \in \mathcal{P}_{t+1}(E')$.

Lemma 5.5.1 *Let t be a nonnegative integer. Let $y \in \text{SA}^t(\widehat{\text{ATSP}}_{\text{DFJ}}(G))$. Suppose that there exists a dipath $Q \subseteq E$ from some vertex q to another vertex p such that $y_e = 1$ for each $e \in Q$. Let V_Q denote the set of internal vertices of the dipath Q , and let $G' = G(V - V_Q) = G - V_Q$. Then,*

$$y|_{E(G')} \in \text{SA}^t(\widehat{\text{PATSP}}_{p,q}(G')).$$

Proof. Let $V' = V - V_Q$ and let $E' = E(G')$, i.e., $G' = (V', E')$. The proof is by induction on t . Denote $y|_{E'}$ by y' for short. Clearly, $y'_\emptyset = 1$. Thus, we only need to prove $y' \in \text{SA}^t(\text{PATSP}_{p,q}(G'))$.

Base case: $t = 0$. Let z be the subvector of y on the singleton sets $\{e_i\}$, and let z' be the subvector of y' on the singleton sets.

We have to prove that z' is a feasible solution of $\widehat{\text{PATSP}}_{p,q}(G')$. It is easy to see that z' is in $[0, 1]^{E'}$ and it satisfies the degree constraints. Thus, we are left with the verification of the cut constraints. Observe that each positive edge (on which z is positive) of G with its head (tail) in V_Q has its tail (head) in $V_Q + q$ ($V_Q + p$). Let $\emptyset \neq U \subseteq V'$. If $U \subseteq V' - \{q\}$, then observe that every edge in $\delta_G^{\text{out}}(U)$ has its head in $V - V_Q - U = V' - U$, hence, we have $z'(\delta_{G'}^{\text{out}}(U)) = z(\delta_G^{\text{out}}(U)) \geq 1$. Similarly, if $U \subseteq V' - \{p\}$, then we have $z'(\delta_{G'}^{\text{in}}(U)) = z(\delta_G^{\text{in}}(U)) \geq 1$; the equation holds because every edge in $\delta_G^{\text{in}}(U)$ has its tail in $V - V_Q - U = V' - U$.

Induction Step: For $t \geq 0$, we know $y' \in \text{SA}^{t+1}(\text{PATSP}_{p,q}(G'))$ if and only if for any $e \in E'$,

$$\begin{aligned} e * y' &\in \text{SA}^t(\text{PATSP}_{p,q}(G')), \\ y' - e * y' &\in \text{SA}^t(\text{PATSP}_{p,q}(G')). \end{aligned}$$

Since y is a feasible solution in $\text{SA}^{t+1}(\text{ATSP}_{DFJ}(G))$, we have

$$\begin{aligned} e * y &\in \text{SA}^t(\text{ATSP}_{DFJ}(G)), \\ y - e * y &\in \text{SA}^t(\text{ATSP}_{DFJ}(G)). \end{aligned}$$

Note that $e \in E'$. For any $S \subseteq E'$ such that $|S| \leq t + 1$, we have $(e * y')_S = y'_{S \cup \{e\}} = y_{S \cup \{e\}} = (e * y)_S$. Thus, $e * y' = (e * y)|_{E'}$. Similarly, we have $y' - e * y' = (y - e * y)|_{E'}$. For any $e^i \in Q$, since $y_{e_i} = 1$, we have $y_{\{e, e_i\}} = y_e$ (by the definition of the SA system), hence, we have

$$(e * y)_{\{e^i\}} = y_{\{e, e_i\}} = y_e = (e * y)_\emptyset.$$

Similarly,

$$(y - e * y)_{\{e^i\}} = y_{e^i} - y_{\{e, e_i\}} = 1 - y_e = (y - e * y)_\emptyset.$$

Case 1: $(e * y)_\emptyset = 0$. In this case, all items in $e * y$ are zero. Thus, $e * y' \in \text{SA}^t(\text{PATSP}_{p,q}(G'))$.

Case 2: $(e * y)_\emptyset > 0$. In this case, we consider $\frac{e * y}{(e * y)_\emptyset}$. Note that $(\frac{e * y}{(e * y)_\emptyset})_{\{e^i\}} = 1$ for any $e^i \in Q$ and $\frac{e * y}{(e * y)_\emptyset} \in \text{SA}^t(\text{ATSP}_{DFJ}(G))$ with value 1 at the item indexed by \emptyset . By the inductive hypothesis, we have $\frac{e * y}{(e * y)_\emptyset}|_{E'} \in \text{SA}^t(\text{PATSP}_{p,q}(G'))$, i.e., $\frac{e * y'}{(e * y')_\emptyset} \in \text{SA}^t(\text{PATSP}_{p,q}(G'))$. Thus, $e * y' \in \text{SA}^t(\text{PATSP}_{p,q}(G'))$.

Similarly, we have $y' - e * y' \in \text{SA}^t(\widehat{\text{PATSP}}_{p,q}(G'))$. This completes the proof. \square

From the last section, we know that $y^{\emptyset,t}$ (Definition 5.4.4) is in $\text{SA}^t(\text{ATSP}_{DFJ}(G))$, where G is defined in Figure 5.5; note that G is obtained from the digraph and the good decomposition given in Figure 5.1. The solid edges in G have cost 1 and the dashed edges in G have cost 0.

Let q be the right-most vertex in the second row (incident to two dashed edges), let p be the left-most vertex in the second row (incident to two dashed edges), and let Q be the dipath of solid edges from q to p . By the definition of $y^{\emptyset,t}$, we have $y_{e_i}^{\emptyset,t} = 1$ for each $e_i \in Q$. Let $G' = G(V')$ where $V' = V - V_Q$ where V_Q is the set of internal vertices of the dipath Q . The next result is a direct corollary of Lemma 5.5.1.

Corollary 5.5.2 *We have*

$$y^{\emptyset,t}|_{E(G')} \in \text{SA}^t(\widehat{\text{PATSP}}_{p,q}(G')), \quad \forall t \geq 0.$$

The proof of the next lemma follows from arguments similar to those in the proof of Theorem 5.4.11.

Lemma 5.5.3 *The minimum cost of a (p, q) -Eulerian subdigraph of G' is $\geq 3\ell$, where ℓ is the number of edges in the middle path in G .*

Theorem 5.5.4 *Let t be a nonnegative integer, and let $\epsilon \in \mathbb{R}$ satisfy $0 < \epsilon \ll 1$. There exists a digraph on $\nu = \nu(t, \epsilon) = \Theta(t/\epsilon)$ vertices such that the integrality ratio for the level- t tightening of the standard relaxation (for the p - q path ATSP, by the **Sherali-Adams** system) is $\geq 1 + \frac{2-\epsilon}{3t+4}$.*

Proof. Given t and ϵ , we fix $\ell = 2(3t + 4)/\epsilon$. Consider the metric completion H of G' . By Section 5.2.2, we can extend the feasible solution from Corollary 5.5.2 to a feasible solution to $\text{SA}^t(\widehat{\text{PATSP}}_{p,q}(H))$. This gives an upper bound on the optimal value of a fractional feasible solution to $\text{SA}^t(\widehat{\text{PATSP}}_{p,q}(H))$. On the other hand, Lemma 5.5.3 gives a lower bound on the optimal value of an integral solution. Thus, the integrality ratio is at least

$$\frac{3\ell}{\frac{t+1}{t+2}2\ell + \ell + 2} \geq 1 + \frac{2}{3t+4} - \frac{2}{\ell} \geq 1 + \frac{2-\epsilon}{3t+4}.$$

\square

Chapter 6

Approximating (Unweighted) Tree Augmentation via Lasserre System, Part I: Stemless TAP

In the weighted *Tree Augmentation Problem* we are given a connected, undirected graph G with non-negative costs on the edges, together with a spanning tree T of G .¹ Let \hat{E}_T be the set of edges of T . The goal is to find a set of edges, $F \subseteq E(G) - \hat{E}_T$, of minimum cost such that the graph $(V, \hat{E}_T \cup F)$ is 2-edge connected. By a link we mean an element of $E(G) - \hat{E}_T$; thus, a link is an edge of G that can be used to augment T . We say that a link uw covers an edge $\hat{e} \in \hat{E}_T$ (a tree-edge) if the graph $T + uw - \hat{e} = (V, \hat{E}_T \cup \{uw\} - \{\hat{e}\})$ is connected. We say that a set of links F covers the tree T if every edge of T is covered by at least one link of F ; it can be seen that F covers T iff the graph $(V, \hat{E}_T \cup F)$ is 2-edge connected. Thus, the goal is to compute a set of links of minimum cost that covers T .

The weighted Tree Augmentation Problem was first studied by Frederickson and Jaja in 1981 [32]. They show that the problem is **NP-hard**, and they present a 2-approximation algorithm. Subsequently, it has been proved that even the unweighted Tree Augmentation Problem is **APX-hard**, see [43, Section 4]; thus the unweighted problem has no **PTAS** assuming $P \neq NP$.

There have been some important advances on this problem for the corresponding *unweighted* (i.e., uniform weight) problem. Recall that we use the abbreviation TAP for the *unweighted* Tree Augmentation Problem. Nagamochi [55] presented the first algorithm for TAP that improved on the approximation guarantee of 2; the approximation guarantee is ≈ 1.875 . Subsequently, Even

¹The results in this chapter are based on a joint-authored paper [12].

et al. [29] built on the ideas and techniques initiated by Nagamochi and presented an elegant algorithm and analysis that achieves an approximation guarantee of 1.8.

The threshold of 1.8 is a natural barrier due to a particular type of subtree configuration, a so-called “stem”, that occurs in instances of TAP (see Section 6.2 for definitions).

To improve on the approximation guarantee of 1.8, both the algorithm and analysis have to be refined to handle stems. This introduces significant new complications. In a conference publication from 2001, Even et al. [28] reported the first 1.5-approximation algorithm for TAP. Very recently, Kortsarz and Nutov finalized the journal version of this result [44].

There are several other papers on TAP, see e.g., [16, 51, 21, 30, 35], but we do not discuss them since either they are not directly relevant to our discussion or they are unpublished manuscripts.

Linear programming relaxations for the weighted version of TAP have been studied for many years. There is a covering LP (relaxation); see Section 2.1.3. It is well known that the integrality ratio of this LP is ≤ 2 ; this can be deduced from Jain’s result [40]. A lower bound of 1.5 on the integrality ratio is known [17]; in fact, the construction for the lower bound uses uniform weights for the edges in $E(G) - \hat{E}_T$, hence, the lower bound applies for TAP.

6.1 Our results and techniques

In this chapter, our main result is to apply the **Lasserre** system to a special case of TAP, and to derive some properties of the feasible solutions that are then used to analyze the integrality ratio and approximation guarantee. A stem is a node s incident to three edges of T that satisfies some additional conditions (see Section 6.2 for formal definitions, and see Figure 6.2 for an illustration). We use *stemless TAP* to refer to the special case of TAP where the instance is such that the tree T has no stem. In this chapter, we prove an approximation guarantee of $1.5 + \epsilon$ relative to a solution of the **Lasserre** system for stemless TAP, for any $\epsilon > 0$.

The **Las** system applies to an initial LP (relaxation), and it derives a sequence of tightenings of the initial LP. A key “decomposition theorem” (see Theorem 6.4.1, [58, 41]) asserts that a feasible solution at level t can be written as a convex combination of feasible solutions at a lower level such that all of these lower-level solutions y are “locally integral.” Here, “locally integral” means that there is a specified subset $J \subseteq E(G) - \hat{E}_T$, such that the solution y takes only zero or one values on this subset (i.e., $y_e \in \{0, 1\}$, $\forall e \in J$). A key point is that the difference in levels (between the level t of the given feasible solution and the level of the lower-level “locally integral” solutions) does *not* depend on the size of J , rather, it depends on the

following “combinatorial parameter” determined by J . Suppose that there exists a constant k such that every feasible solution x of the initial LP has $\leq k$ entries in J that have value one, i.e., $|\{e \in J : x_e = 1\}| \leq k$ for every feasible solution x of the initial LP. Then for any $t > k$, a feasible solution at level t can be written as a convex combination of “locally integral” feasible solutions at level $(t - k)$. This property does not hold for other weaker Lift-and-Project systems such as the Lovász-Schrijver system or the Sherali-Adams system.

We formulate an initial LP that is a tightening of the covering LP for TAP; see [LP7](#) in Section [6.3](#). For this purpose, we introduce the notion of overlapping pairs, and we add a family of constraints on overlapping pairs to our initial LP; see Proposition [6.3.1](#) in Section [6.3](#). This (together with the decomposition theorem) turns out to be the key for proving properties of feasible solutions to the Las system.

The analysis of our algorithm is based on a potential function. Our potential function is derived from the Las tightening of the initial LP. In contrast, the previous literature uses potential functions that are derived from combinatorial lower bounds. We present an example showing that our potential function is not valid for our initial LP; in other words, the Las tightening is essential for one part of our analysis.

Our algorithm is “combinatorial” and we do *not* need to solve the initial LP nor its Las tightening to run the algorithm (but, the analysis of the algorithm relies on the Las tightening). Our algorithm may be viewed as a variant of the algorithm of [\[29, Section 3.4\]](#); see Section [6.6](#) for details.

The algorithm is a greedy-type iterative algorithm that makes a leaves-to-root scan over the tree T and (incrementally) constructs a set of links F that covers T . The algorithm starts with $F := \emptyset$, at each major step it adds one or more links to F (it never removes links from F), and at termination, it outputs a set of links F that covers T such that $|F| \leq$ the potential function. The algorithm incurs a cost of one unit for each link added to F . The key to the analysis is to show that for each major step, the cost incurred (i.e., one plus the number of links added to F) is compensated by a part of the potential function; see Section [6.6](#) for details.

It is possible that the naive algorithm gets “stuck.” But, in this scenario, we can prove that there exists a small combinatorial obstruction. The algorithm can be modified for this scenario. The modified algorithm finds each occurrence of the small combinatorial obstruction in polynomial time, and then handles all of these occurrences in an appropriate way; see Section [6.7.2](#) for details; this part is similar to [\[29, Section 4.3\]](#).

Informally speaking, our analysis in Section [6.7](#) asserts the following:

if the naive algorithm gets “stuck” then there exists a small combinatorial obstruction, a so-called deficient 3-leaf tree, see Theorems [6.7.3](#), [6.7.4](#).

This assertion is the key to this chapter; it turns out that the algorithmic aspects as well as the analysis of the integrality ratio and approximation guarantee are straightforward consequences. Our analysis in Section 6.7 makes essential use of the **Las** system and the decomposition theorem; see Figure 6.1.

Finally, we present an example of stemless TAP such that the approximation guarantee of $\frac{3}{2}$ is tight for the algorithm.

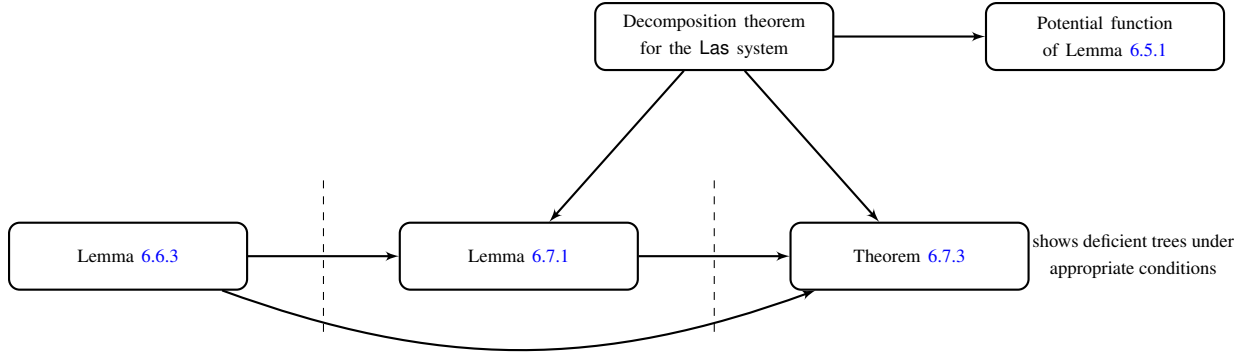


Figure 6.1: An illustration of the role of the decomposition theorem for the **Las** system (Theorem 6.4.1) in our analysis. The analysis for Theorem 6.7.3 consists of three blocks: low-level assertions maintained by the algorithm, intermediate-level results (on credits and structural properties), and the high-level analysis (proof of Theorem 6.7.3); the three blocks are shown from left to right in the figure. The decomposition theorem pertains to the two right-most blocks. Also, we use the decomposition theorem to derive our potential function (Lemma 6.5.1).

In Chapter 7, we extend the methods of this chapter to prove the same approximation guarantee for (general) TAP. Chapter 7 follows the same outline as this chapter. Moreover, the initial LP and **Las** tightening are the same for both chapters. Our motivation for writing this chapter is to give an accessible presentation of the algorithmic ideas and “flow of arguments” used in Chapter 7.

An outline of this chapter is as follows. Section 6.2 has definitions and notation. We adopt the notation and terms of Even et al. [29], where possible; this will aid readers familiar with that paper. Section 6.3 presents the initial LP, while Section 6.4 discusses the **Las** tightening of the initial LP, and proves some basic properties and inequalities. Section 6.5 derives our potential function, based on a solution of the **Las** tightening; this section also has an example showing that our potential function is not valid for the initial LP. Section 6.6 presents the algorithm and the “credit scheme” used by the algorithm. The most important component of this chapter is Section 6.7; this section proves the key theorem on deficient trees (Theorem 6.7.3); this section

also presents and proves the last piece of the algorithm, namely the handling of deficient trees. Section 6.8 presents an example of stemless TAP such that the approximation guarantee of $\frac{3}{2}$ is tight for the algorithm.

6.2 Preliminaries and notation

This section presents definitions and notation.

Standard notation including tree T , link set E

Let $G = (V, E(G))$ be a connected, undirected graph, and let $T = (V, \hat{E}_T)$ be a spanning tree of G . We assume that $|V| \geq 2$. Recall some notation from Section 2.1.3. By a *tree-edge* we mean an edge of T . Let E denote the edge-set $E(G) - \hat{E}_T$; we call E the *link set* and we call an element $\ell \in E$ a *link*; thus, a link is an edge of G that can be used to augment T . An instance of TAP consists of G and T . We assume that all instances of interest have feasible solutions, that is, we assume that $(V, \hat{E}_T \cup E)$ is 2-edge connected. The goal is to find a minimum-size subset F of E such that augmenting T by F results in a 2-edge connected graph, i.e., the graph $(V, \hat{E}_T \cup F)$ is 2-edge connected.

For two nodes $u, v \in V$, we use $P_{u,v} = P_{v,u}$ to denote the unique path of the tree T between u and v .

For a node $v \in V$, we denote the number of tree-edges incident to v by $\deg_T(v)$. Let uv be a link such that $\deg_T(u) = 1$, $\deg_T(v) = 1$, there exists an internal node s in $P_{u,v}$ such that $\deg_T(s) = 3$, and every other internal node w (if any) in $P_{u,v}$ has $\deg_T(w) = 2$. Then, uv is called a *twin link* and s is called a *stem*.

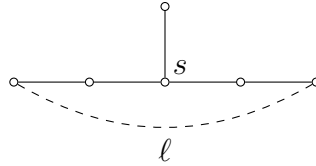


Figure 6.2: Illustration of a stem s and its twin link ℓ . The tree-edges are indicated by solid lines and the twin link ℓ is indicated by dashed lines.

We use *stemless TAP* to refer to the special case of TAP where the instance is such that the tree T has no stems.

For any $U \subseteq V$, we denote the set of links with both ends in U by $E(U)$, and for any two subsets U, W of V , we denote the set of links with one end in U and the other end in W by $E(U, W)$; thus, $E(U, W) := \{uv \in E : u \in U, w \in W\}$.

Recall from Section 2.1.3 that a link uv *covers* a tree-edge \hat{e} if $P_{u,v} \ni \hat{e}$.

For any tree-edge $\hat{e} \in \hat{E}_T$, we use $\delta_E(\hat{e})$ to denote the set of links that cover \hat{e} , thus, $\delta_E(\hat{e}) = \{uv \in E : \hat{e} \in P_{u,v}\}$. For any node $w \in V$, we use $\delta_E(w)$ to denote the set of links incident to w .

Shadows and the shadow-closed property

For two links u_1v_1 and u_2v_2 , if $P_{u_1,v_1} \subseteq P_{u_2,v_2}$, then we call u_1v_1 a *shadow* (or, *sublink*) of u_2v_2 . In particular, if $P_{u_1,v_1} \subsetneq P_{u_2,v_2}$, then we call u_1v_1 a *proper shadow* (or, *proper sublink*) of u_2v_2 .

For each link $uv \in E$, if all sublinks of uv exist in E , then we call E *shadow-closed*. Clearly, if E is not shadow-closed, then we can make it shadow-closed by adding all sublinks of each of the original links. It can be seen that this preserves the optimal value of any instance of TAP.

Observe that a twin link cannot be a proper sublink of another link, because each end u of a twin link has $\deg_T(u) = 1$. Thus, when we add sublinks to E to make it shadow-closed, then none of the added sublinks can be a twin link. Hence, if we start with a stemless instance of TAP, then we do not introduce any stems when we make it shadow-closed.

Following Even et al., see [29, Assumption 2.2], we make the next assumption.

Assumption: E is shadow-closed.

Contractions, compound nodes and original nodes

By an *original node* we mean an element of $V(T)$, and by an *original link* we mean an element of E .

Recall that the algorithm (incrementally) constructs a set of links F such that, at termination, $T + F = (V, \hat{E}_T \cup F)$ is 2-edge connected. Throughout, we use F to denote the current solution of the algorithm; initially, $F := \emptyset$. We use the standard notion of *contracting* a link or a set of links, see [29], [25, Chapter 1]. Throughout, we use T/F (or, $T' := T/F$) to denote the “current tree” obtained by contracting each of the 2-edge connected components of $T + F = (V, \hat{E}_T \cup F)$ to a single node. Each of the contracted nodes of T/F is called a *compound node*, see [29, Section 3.2]; thus, each compound node corresponds to a set of two or more nodes of $V(T)$.

Each of the other (non compound) nodes of T' is called an *original node*. For this paragraph, given an original node v of T , let us use v' to denote the corresponding node of T/F , i.e., if v is an original node of T/F , then we have $v' = v$, otherwise v' denotes the compound node of T/F that contains v . Similarly, let us use $u'w'$ to denote the “image” of an original link $uw \in E$ w.r.t. the current tree T/F . For a set of original links $J \subseteq E$, the image w.r.t. the current tree T/F is $\{u'w' : uw \in J, u' \neq w'\}$. Note that original links that have both ends in the same compound node of T/F are discarded since they are not relevant for the rest of the execution/analysis. When we discuss the algorithm and its analysis (in Sections 6.6, 6.7), we may abuse the notation by not distinguishing between an original link $uw \in E$ and its image $u'w'$ w.r.t. the current tree T/F ; similarly, we may not distinguish between a set of original links and the set of images of those original links.

Root, ancestor, descendant and rooted subtrees

One of the nodes r of T is designated as the root; thus, we have a rooted tree (T, r) .

Let v be a node of T . If a node w belongs to the path $P_{v,r}$, then w is called an *ancestor* of v , and v is called a *descendant* of w . If a descendant w of v is adjacent to v (thus, $w \neq v$), then w is called a *child* of v , and v is called a *parent* of w . Clearly, every node (except r) has a unique parent. If v has no child, then we call v a *leaf*; clearly, if v has no child, then $\deg_T(v) = 1$. Note that r is not a leaf, even if $\deg_T(r) = 1$. Throughout, we use L to denote the set of original leaves; we use \mathcal{R} to denote $V - L$, i.e., the set of original non-leaf nodes.

For any node v , we use T_v to denote the rooted subtree of (T, r) induced by v and its descendants. (Throughout, we view T as an “oriented tree” rooted at r , and we use the term subtree to refer to a rooted subtree.)

We say that a subtree T_v is *covered* by a set of links $J \subseteq E$ if each tree-edge of T_v is covered by some link of J .

Property 6.2.1 *Suppose \bar{T} is a rooted tree. Let \bar{T}_{v_1} and \bar{T}_{v_2} be two (rooted) subtrees of \bar{T} . Then \bar{T}_{v_1} and \bar{T}_{v_2} either share no node or one is contained in the other.*

Proof. Suppose \bar{T}_{v_1} and \bar{T}_{v_2} share a node, say w . Then, both v_1 and v_2 are ancestors of w . This implies that one of v_1 or v_2 must be an ancestor of the other one. Hence, one of the two subtrees $\bar{T}_{v_1}, \bar{T}_{v_2}$ is contained in the other one (possibly, $v_1 = v_2$ and $\bar{T}_{v_1} = \bar{T}_{v_2}$). \square

For any leaf v of T , $up(v)$ denotes a node q in $P_{v,r}$ that is nearest to the root and adjacent to v via a link; clearly, $up(v)$ exists by the assumptions of feasibility and shadow-closure.

Vectors and convex combinations

For any vector $x \in \mathbb{R}^E$, let $\text{ones}(x)$ denote the set of links of x -value one, thus $\text{ones}(x) = \{uv \in E : x_{uv} = 1\}$.

For a vector $x \in \mathbb{R}^E$ and any subset J of E , $x(J)$ denotes $\sum_{e \in J} x_e$, and $x|_J$ denotes the restriction of x to J . Given several vectors v^1, v^2, \dots , we write one of their convex combinations as $\sum_{i \in Z} \lambda_i v^i$; thus, Z is a set of indices, and we have $\lambda_i \geq 0, \forall i \in Z$, and $\sum_{i \in Z} \lambda_i = 1$.

Remark 6.2.2 *For expository reasons, this chapter uses definitions of “stem” and “twin link” that are relaxations of the definitions in Chapter 7. The two definitions in this chapter are independent of the choice of the root r . In Chapter 7, we call a node s of the rooted tree (T, r) a stem if s is not the root r and has exactly two children, the subtree T_s has exactly two leaves, and there exists a link between the two leaves; the leaf-to-leaf link is called a twin link. If s is a stem according to Chapter 7, then it must be a stem according to this chapter, but not vice-versa (a similar statement holds for twin links). Clearly, if an instance of TAP is stemless according to this chapter, then it is stemless according to Chapter 7 as well.*

6.3 The initial LP

This section presents our LP relaxation (LP7) for TAP.

Let u_1v_1 and u_2v_2 be a pair of links. We call it an *overlapping pair* of links if (i) P_{u_1,v_1}, P_{u_2,v_2} have one or more tree-edges in common, and (ii) either an end of u_1v_1 is in P_{u_2,v_2} , or an end of u_2v_2 is in P_{u_1,v_1} . We call a set of links J an *overlapping clique* if every pair of links in J is an overlapping pair.

The following LP with constraints on overlapping pairs gives a relaxation of shadow-closed instances of TAP.

$$\begin{aligned}
 & \text{minimize : } \sum_{uv \in E} x_{uv} && \text{(LP7)} \\
 & \text{subject to : } \sum_{uv \in \delta_E(\hat{e})} x_{uv} \geq 1 && \forall \hat{e} \in \hat{E}_T \\
 & && x_{u_1v_1} + x_{u_2v_2} \leq 1 && \forall \text{ overlapping pairs } u_1v_1, u_2v_2 \in E \\
 & && 1 \geq x_{uv} \geq 0 && \forall uv \in E
 \end{aligned}$$

Note that it tightens the feasible region of the covering LP for TAP (see Section 2.1.3),

$$\min\left\{\sum_{e \in E} x_e : x(\delta_E(\hat{e})) \geq 1, \forall \hat{e} \in \hat{E}_T, 1 \geq x \geq 0\right\},$$

because LP7 has additional constraints for the overlapping pairs. The next result shows that LP7 has the same optimal value as the covering LP, for fractional solutions as well as for integral solutions, provided the instance is shadow-closed.

Proposition 6.3.1 *Consider a shadow-closed instance of TAP. The optimal values of LP7 and the covering LP are the same. Moreover, the best objective value of an integral solution of LP7 is the same as the best objective value of an integral solution of the covering LP.*

Proof. Clearly, any feasible solution of LP7 is also a feasible solution of the covering LP. We claim that there exists an optimal solution of the covering LP that is feasible for LP7. The first statement follows from this claim.

Let x be an optimal solution for the covering LP that minimizes $\sum_{uv \in E} \text{length}(P_{u,v}) \cdot x_{uv}$, where $\text{length}(P_{u,v})$ denotes the number of tree-edges of $P_{u,v}$. We show that x is feasible for LP7. Otherwise, suppose that x violates the constraint for an overlapping pair u_1v_1, u_2v_2 . W.l.o.g., suppose that $u_1 \in V(P_{u_2,v_2})$ and some tree-edge is in both P_{u_1,v_1}, P_{u_2,v_2} . Then, there is a maximal (nonempty) prefix of the edge sequence of P_{u_1,v_1} that is contained in P_{u_2,v_2} ; let us denote this prefix by P_{u_1,u_*} . (Note that the link u_*v_1 is present because the instance is shadow-closed.)

Let α denote the original value of $x_{u_1v_1}$. Then, we replace the value $x_{u_1v_1}$ by $1 - x_{u_2v_2}$, thereby enforcing the constraint for the overlapping pair u_1v_1, u_2v_2 . Moreover, we add $\alpha + x_{u_2v_2} - 1$ to the value of $x_{u_*v_1}$, and if the new value of $x_{u_*v_1}$ exceeds 1, then we replace it by 1. It can be seen that this preserves the constraints of the covering LP. This procedure decreases the value $\sum_{uv \in E} \text{length}(P_{u,v}) \cdot x_{uv}$ but does not increase the objective value $\sum_{uv \in E} x_{uv}$. This contradicts the assumption that x is an optimal solution for the covering LP that minimizes $\sum_{uv \in E} \text{length}(P_{u,v}) \cdot x_{uv}$.

The last part (on integral solutions) follows from similar arguments, because the above procedure maintains integrality. \square

6.4 Lasserre tightening and its properties

In this section, we discuss the Las tightening of the initial LP, and proves some basic properties and inequalities. The lemmas in this section, namely Lemmas 6.4.2-6.4.4, apply for all shadow-closed instances of TAP; we do not make use of the stemless property.

Consider our LP relaxation (LP7) for TAP, and let $\text{Las}_{proj}^t(\text{LP7})$ denote the projection (on the subspace \mathbb{R}^E indexed by the singleton) of the level t tightening of LP7 by the LAS system (see the definition of Las_{proj}^t in Section 2.2).

Rothvoß, see [58, Theorem 2], formulated the following decomposition theorem for the LAS system, based on an earlier decomposition theorem due to Karlin, Mathieu and Nguyen [41]. (We use this particular formulation and not the original statement of [41]; hence, we reference both [41] and [58].)

Theorem 6.4.1 *Let $J \subseteq E$. Let k be a positive integer such that $|\text{ones}(x) \cap J| \leq k$ for every feasible solution x of LP7. Then for every feasible solution $y \in \text{Las}_{proj}^t(\text{LP7})$, where $t \geq k + 1$, y can be written as a convex combination: $y = \sum_{i \in Z} \lambda_i x^i$ such that x^i is in $\text{Las}_{proj}^{t-k}(\text{LP7})$ and $x^i|_J$ is integral (i.e., x_{uv}^i is integral for each $uv \in J$), for all $i \in Z$.*

Lemma 6.4.2 *Let $J \subseteq E$ be an overlapping clique. For every feasible solution x of LP7, we have $|\text{ones}(x) \cap J| \leq 1$. Furthermore, for every level $t \geq 2$ and every feasible solution y of $\text{Las}_{proj}^t(\text{LP7})$, we have $y(J) \leq 1$.*

Proof. Let x be a feasible solution of LP7. Then the overlapping constraints in LP7 imply that $|\text{ones}(x) \cap J| < 2$. To see this, suppose that $|\text{ones}(x) \cap J| \geq 2$. Then there exists a pair of links $u_1v_1, u_2v_2 \in J$, with $x_{u_1v_1} = x_{u_2v_2} = 1$; thus, u_1v_1, u_2v_2 is an overlapping pair whose associated constraint in LP7 is violated.

By Theorem 6.4.1, y can be written as a convex combination: $y = \sum_{i \in Z} \lambda_i x^i$ such that x^i is in $\text{Las}_{proj}^1(\text{LP7})$ and $x^i|_J$ is integral for each $i \in Z$. Hence, $x^i(J) \leq 1$ for each $i \in Z$, because $|\text{ones}(x^i) \cap J| \leq 1$ and $x^i|_J$ is integral. Consequently, the convex combination y of $x^i, i \in Z$, satisfies $y(J) \leq 1$. \square

Lemma 6.4.3 *Let w be a leaf of T . Let \hat{e} be a tree-edge with ends u and v such that v is a child of u , w is a descendant of v (possibly $w = v$), and every internal node (if any) of the path $P_{w,u}$ has exactly one child. Then, $\delta_E(\hat{e})$ is an overlapping clique. In particular, $\delta_E(w)$ is an overlapping clique.*

Moreover, we have $y(\delta_E(w)) = 1$ for any feasible solution $y \in \text{Las}_{proj}^t(\text{LP7})$ where $t \geq 2$.

Proof. Consider any two links $f_1q_1, f_2q_2 \in \delta_E(\hat{e})$. Clearly, each of the links f_1q_1, f_2q_2 must have an end in $P_{w,v}$. Suppose that q_1, q_2 are the ends in $P_{w,v}$, and w.l.o.g., assume that q_1 is an ancestor of q_2 . Then, observe that f_1q_1, f_2q_2 is an overlapping pair, because q_1 is in P_{f_2,q_2} and

the tree-edge \hat{e} is in both P_{f_1, q_1} and P_{f_2, q_2} (see Figure 6.3). Hence, the set of links covering \hat{e} is an overlapping clique.

If we take \hat{e} to be the unique tree-edge incident to the leaf w , then it can be seen that $\delta_E(w) = \delta_E(\hat{e})$ is an overlapping clique. Moreover, by Lemma 6.4.2, $y(\delta_E(w)) \leq 1$, whereas the constraints of LP7 imply that $y(\delta_E(w)) = y(\delta_E(\hat{e})) \geq 1$. Therefore, $y(\delta_E(w)) = 1$. \square

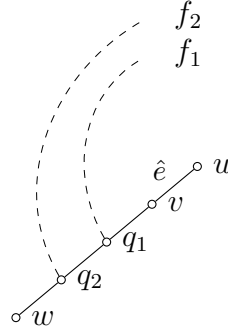


Figure 6.3: Illustration of the proof of Lemma 6.4.3. The solid lines are tree-edges and the dashed lines are links.

Recall that the matching polytope of the subgraph induced by the leaves, $G(L) = (L, E(L))$ is given by the following constraints:

$$\begin{aligned} x(\delta_{E(L)}(v)) &\leq 1 & \forall v \in L \\ x(E(W)) &\leq \frac{|W| - 1}{2} & \forall W \subseteq L, |W| \text{ odd} \\ x &\geq 0 \end{aligned}$$

The next result is essentially the result on the matching polytope from the survey of Rothvoß, see [59, Lemma 13, Sec 3.3], translated to our setting.

Lemma 6.4.4 *Let $\epsilon > 0$, and let $t \geq \frac{1}{2\epsilon} + 1$. Suppose that $y \in \text{Las}_{\text{proj}}^t(\text{LP7})$ is the projection of a feasible solution of the level t of the **Las** system. Then, $\frac{y|_{E(L)}}{1+\epsilon}$ is in the matching polytope of $G(L) = (L, E(L))$.*

Proof. By Lemma 6.4.3, we have $y(\delta_{E(L)}(v)) \leq y(\delta_E(v)) \leq 1, \forall v \in L$. Hence, for any set $W \subseteq L$, we have $y(E(W)) \leq |W|/2$. For “large” odd sets $W \subseteq L$, we will show that $y(E(W)) \leq |W|/2$ implies that $\frac{y(E(W))}{1+\epsilon} \leq \frac{|W|-1}{2}$, whereas, for non-large odd sets W , we deduce $y(E(W)) \leq \frac{|W|-1}{2}$ by the decomposition theorem (Theorem 6.4.1) and local integrality on $E(W)$.

First, consider odd sets $W \subseteq L$ with $|W| > \frac{1}{\epsilon} + 1$. Clearly, we have $y(E(W)) \leq \frac{|W|}{2}$ since $y(\delta_{E(L)}(v)) \leq 1, \forall v \in L$. Also, observe that $\frac{|W|}{2} = \frac{|W|-1}{2}(1 + \frac{1}{|W|-1}) < \frac{|W|-1}{2}(1 + \frac{1}{\frac{1}{\epsilon}+1-1}) = \frac{|W|-1}{2}(1 + \epsilon)$. Hence, $y(E(W)) \leq \frac{|W|-1}{2}(1 + \epsilon)$.

Now, consider odd sets $W \subseteq L$ with $|W| \leq \frac{1}{\epsilon} + 1$. We apply the decomposition theorem, Theorem 6.4.1. Note that for any feasible solution x of LP7, by Lemmas 6.4.2, 6.4.3, we have $|\text{ones}(x) \cap E(W)| \leq \frac{|W|-1}{2} \leq \frac{1}{2\epsilon} \leq t-1$. Since $y \in \text{Las}_{proj}^t(\text{LP7})$, we have y can be written as a convex combination: $y = \sum_{i \in Z} \lambda_i x^i$ such that x^i is in $\text{Las}_{proj}^1(\text{LP7})$ and $x^i|_{E(W)}$ is integral. Note that $\delta_E(w)$ is an overlapping clique by Lemma 6.4.3 for every $w \in W$. Hence, for each $i \in Z$, $x^i(E(W)) \leq \frac{|W|-1}{2}$. Consequently, $y(E(W)) \leq \frac{|W|-1}{2}$. This completes the proof. \square

6.5 Potential function for stemless TAP

This section presents the potential function that is used in our analysis.

Let M denote a maximum matching of $(L, E(L))$; thus, M is a maximum matching of the leaf-to-leaf links. By an M -link we mean a link that is in M . Let U denote the set of M -exposed leaf nodes, that is, the set of leaves that are not covered by M .

We will often refer to M and U in the rest of this chapter; these are key items for the algorithm (Section 6.6) and its analysis (Section 6.7).

Recall that \mathcal{R} is the set of non-leaf nodes, i.e., $\mathcal{R} = V - L$.

Lemma 6.5.1 *Let $\epsilon > 0$ be a constant, and let $t \geq 1 + \frac{1}{2\epsilon}$. Let $y \in \mathbb{R}^E$ be the projection of a feasible solution of the level t of the **Las** system, i.e., $y \in \text{Las}_{proj}^t(\text{LP7})$. Then*

$$\left(\frac{3}{2} + \epsilon\right)y(E) \geq |U| + \frac{3}{2}|M| + \frac{1}{2} \sum_{v \in \mathcal{R}} y(\delta_E(v)).$$

Proof.

$$\begin{aligned} \frac{3}{2}y(E) &= \frac{3}{2}y(E(L)) + \frac{3}{2}y(E(L, \mathcal{R})) + \frac{3}{2}y(E(\mathcal{R})) \\ &\geq \left(2y(E(L)) - \frac{1}{2}y(E(L))\right) + \left(y(E(L, \mathcal{R})) + \frac{1}{2}y(E(L, \mathcal{R}))\right) + y(E(\mathcal{R})) \\ &= \left(2y(E(L)) + y(E(L, \mathcal{R}))\right) - \frac{1}{2}y(E(L)) + \left(\frac{1}{2}y(E(L, \mathcal{R})) + y(E(\mathcal{R}))\right) \end{aligned}$$

$$\begin{aligned}
&= \sum_{v \in L} y(\delta_E(v)) - \frac{1}{2}y(E(L)) + \frac{1}{2} \sum_{v \in \mathcal{R}} y(\delta_E(v)) \\
&\stackrel{(1)}{\geq} |L| - \frac{1}{2}(1 + \epsilon)|M| + \frac{1}{2} \sum_{v \in \mathcal{R}} y(\delta_E(v)) \\
&\stackrel{(2)}{=} |U| + \left(\frac{3}{2} - \frac{\epsilon}{2}\right)|M| + \frac{1}{2} \sum_{v \in \mathcal{R}} y(\delta_E(v)),
\end{aligned}$$

where (1) follows from two facts that $y(\delta_E(v)) = 1$ for any $v \in L$ by Lemmas 6.4.3 and $\frac{y(E(L))}{1+\epsilon} \leq |M|$ by Lemma 6.4.4, and (2) follows from the observation that $|L| = |U| + 2|M|$. Note that $|M| \leq \frac{1}{2}|L| = \frac{1}{2} \sum_{v \in L} y(\delta_E(v)) \leq y(E)$, hence, $-\frac{\epsilon}{2}|M| \geq -\epsilon y(E)$. Thus, we have $(\frac{3}{2} + \epsilon)y(E) \geq |U| + \frac{3}{2}|M| + \frac{1}{2} \sum_{v \in \mathcal{R}} y(\delta_E(v))$. \square

Lemma 6.5.1 states a key inequality for the level- t tightening of LP7. But, the following example shows that this key inequality does not hold for all feasible solutions of LP7 (without tightening by the LAS system). The example has a parameter k . We start with a path of length $k - 1$, then attach a claw (a copy of $K_{1,3}$) at each node of the path. This gives the tree T . The link set E consists of a link from one end of the path to the other end, and the three links connecting every pair of leaves in each copy of the claw. Note that all the sublinks of these links are also contained in E (see Figure 6.4).

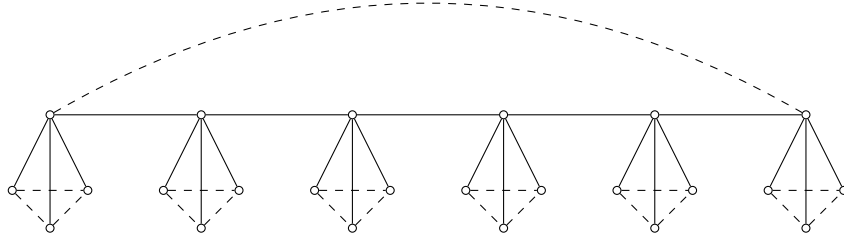


Figure 6.4: Instance for $k = 6$. The tree-edges of T are indicated by solid lines, and the maximal links in E are indicated by dashed lines.

Define a feasible solution x of LP7 as follows: the link from one end of the path to the other end gets value 1, every link connecting a pair of leaves in each claw gets value $\frac{1}{2}$, and all other links get value 0. It is not hard to see that x is feasible for LP7 and that $x(E) = \frac{3}{2}k + 1$. (Note that the optimal value of LP7 may be $< x(E)$, but our arguments do not use the optimal value.)

If we pick any node on the path to be the root node, then any maximum matching $M \subseteq E(L)$ has size k and there are k M -exposed leaves. Thus, $\frac{3}{2}|M| + |U| = \frac{5}{2}k$, and this quantity is larger

than $(\frac{3}{2} + \epsilon)x(E) = (\frac{9}{4} + \frac{3}{2}\epsilon)k + (\frac{3}{2} + \epsilon)$ for any $\epsilon < \frac{1}{6}$ and for sufficiently large k . Thus, the inequality stated in Lemma 6.5.1 does not hold for x .

6.6 Algorithm

This section presents the details of our algorithm and its analysis following the overview given in Section 6.1. The working of the algorithm is illustrated below, see Figure 6.5.

We first state our main result for (unweighted) stemless TAP:

Theorem 6.6.1 *Consider an instance of stemless TAP. Let $\epsilon > 0$ be a constant, and let $t \geq \max\{3, \frac{1}{2\epsilon} + 1\}$. The integrality ratio of $\text{Las}_{proj}^t(\text{LP7})$ is $\leq \frac{3}{2} + \epsilon$. Moreover, there is a polynomial-time algorithm for finding a feasible solution of TAP of size $\leq (\frac{3}{2} + \epsilon)y(E)$, where y is an optimal solution of $\text{Las}_{proj}^t(\text{LP7})$.*

For the rest of this chapter, we fix $y \in \mathbb{R}^E$ to be an optimal solution in $\text{Las}_{proj}^t(\text{LP7})$, where $t \geq \max\{3, 1 + \frac{1}{2\epsilon}\}$, where $\epsilon > 0$ is any (small) constant. We take the right-hand side of the inequality in Lemma 6.5.1 to be our potential function. Thus, our potential function is $\leq (\frac{3}{2} + \epsilon)y(E)$. Our goal is to present an algorithm that finds a set of links F that covers T such that $|F|$ is \leq our potential function; then, it will follow that $|F|$ is within a factor of $(\frac{3}{2} + \epsilon)$ of optimal.

The purpose of the potential function is to provide “credits” to the algorithm. In Section 6.6.2, we distribute the credit (i.e., the potential function) among the nodes and links.

Recall from Section 6.1 that the algorithm maintains a set of links F and a current tree $T' := T/F$. Initially, we have $F := \emptyset$, and $T' := T$.

Consider $T' := T/F$ and the addition of a single link $\ell = vw$ to F . The new tree $T/(F \cup \{\ell\})$ can be obtained by contracting the unique path of T' between v and w , $P'_{v,w}$, to a single compound node. Besides adding a single link in a major step, the algorithm may add a set of links $\{\ell_1 = v_1w_1, \ell_2 = v_2w_2, \dots, \ell_k = v_kw_k\}$ such that the union of the paths of T' corresponding to $\ell_1, \ell_2, \dots, \ell_k$ forms a single connected component, i.e., $\bigcup_{i=1}^k P'_{v_i, w_i}$ is a connected subgraph of T' . Again, the new tree can be obtained by contracting all of these paths into a single compound node.

The algorithm repeatedly finds a set of links $F^{iter} \subseteq E - F$ such that the contraction of F^{iter} in the current tree results in a single new compound node, and moreover, the credit available from the contraction of F^{iter} is $\geq |F^{iter}| + 1$ (the credit 1 is for the new compound node; the details

are discussed below). We add F^{iter} to F , and obtain an updated current tree T' . The algorithm repeats this until T' is a single node, that is, until $T + F$ is 2-edge connected.

There are two types of link sets that get contracted by a major step. The first type is a singleton set, i.e., the major step adds one link to F . The second type is defined via the notion of a semiclosed tree. This is discussed in the next subsection.

6.6.1 Semiclosed trees

We start by defining the key notion of a semiclosed tree w.r.t. an *arbitrary* matching of the leaf-to-leaf links. This notion is due to Even et al., based on earlier work by Nagamochi [55]; also, see [29, Definition 2.3].

Let T'_v be a rooted subtree of the current tree $T' = T/F$. Let \bar{M} be an arbitrary matching of the leaf-to-leaf links. T'_v is called *semiclosed* w.r.t. \bar{M} if the following conditions hold:

- (i) Each link in \bar{M} either has both ends in T'_v or has no end in T'_v .
- (ii) Every link incident to an \bar{M} -exposed leaf of T'_v has both ends in T'_v . (Thus, if $T'_v \neq T'$, then none of the links covering the tree-edge between v and its parent is incident to an \bar{M} -exposed leaf of T'_v .)

Let $\bar{M}(T'_v)$ denote the set of links in \bar{M} that have both ends in T'_v .

We define

$$\Gamma(\bar{M}, T'_v) := \bar{M}(T'_v) \cup \{up(w)w : w \text{ is an } \bar{M}\text{-exposed leaf of } T'_v\};$$

thus, we associate a “basic link set” with the pair \bar{M}, T'_v . In general, the “basic link set” may not be a cover of T'_v .

By a *minimally semiclosed tree* T'_v we mean that T'_v is semiclosed but none of the proper rooted subtrees of T'_v is semiclosed.

Lemma 6.6.2 (Even et al. [29]) *Let T'_v be a minimally semiclosed tree w.r.t. \bar{M} . Then $\Gamma(\bar{M}, T'_v)$ covers all the tree-edges of T'_v .*

Proof. Suppose that some tree-edge $\hat{e} = pq$ is not covered by $\Gamma(\bar{M}, T'_v)$, where q is a child of p . Observe that pq cannot be incident to a leaf of T'_v since $\Gamma(\bar{M}, T'_v)$ has a link incident to each leaf of T'_v . Then, consider the rooted subtree T'_q . It can be seen that T'_q is a semiclosed tree w.r.t. \bar{M} , otherwise, $\Gamma(\bar{M}, T'_v)$ would cover $\hat{e} = pq$. This contradicts the fact that T'_v is minimally semiclosed. \square

6.6.2 Credit assignment

Recall that M is a maximum matching of $(L, E(L))$, U is the set of M -exposed leaf nodes and \mathcal{R} is the set of (original) non-leaf nodes in T .

We start with the credit given by the potential function of Lemma 6.5.1, and we maintain the following assignment of credits to the nodes of $T' := T/F$ and the links of M :

- every M -exposed original leaf has one credit,
- every compound node has one credit,
- every (original) node $v \in \mathcal{R}$ is assigned $\frac{1}{2}y(\delta_E(v))$ credit,
- every M -link has $\frac{3}{2}$ credit, and
- the (original) root r has one credit.

It can be seen that the potential function of Lemma 6.5.1 suffices for assigning credits to the initial tree $T' := T$, except for the unit credit for the root r . When the algorithm terminates, the tree T' becomes a single compound node with one credit. But, this credit will not be used any more, and thus we have a surplus of one credit. We assign this surplus credit to the root r at initialization.

We mention that the nodes or links that get contracted into a compound node are no longer relevant for the algorithm or the analysis. In particular, the credit (if any) of such nodes or links may be used at the step when they get contracted into a compound node, but after that step, any remaining credit of such nodes or links is not used at all. We take the credit of a link ℓ w.r.t. the current tree T' to be the credit of the original link corresponding to ℓ .

6.6.3 Simple contractions and assertions on M

Let $\ell = uw$ be a link, where u, w are nodes of the current tree T' , and let $P'_{u,w}$ denote the path of T' between u and w . We call $\ell = uw$ a *good link* if the sum of the following items is ≥ 2 : (i) the credit of $\ell = uw$, (ii) the number of compound nodes in $P'_{u,w}$, (iii) the number of M -exposed original leaves in $P'_{u,w}$, and (iv) 1 if the root r is an original node of $P'_{u,w}$. In other words, if we take the credits associated with y to be “fractional credits,” then uw is a good link if the “non fractional credits” associated with uw and the nodes of $P'_{u,w}$ is ≥ 2 .

We define a *simple contraction* to be one of the following types of single-link contractions.

- For the current tree, consider a leaf-to-leaf link uw such that each end owns one credit; thus each of u, w is either a compound leaf node or an original leaf node that is M -exposed. Observe that uw is a good link.
- For the current tree, consider an M -link uw such that the path between u and w in the current tree contains at least one compound node. Again, note that uw is a good link.

Lemma 6.6.3 (Assertions on M) *Suppose that no simple contractions are applicable. Then*

- (1) *For every M -link uw , every node in the path between u and w in T' is an original node. In particular, in T' , both ends of each M -link are original leaf nodes.*
- (2) *There exist no links between M -exposed leaves.*

6.6.4 Good semiclosed trees

For the rest of this chapter, unless mentioned otherwise, a semiclosed tree means a tree that is semiclosed w.r.t. the matching M .

Recall that a semiclosed tree is defined w.r.t. an arbitrary matching of the leaf-to-leaf links. We chose $M \subseteq E$ to be a maximum matching of the leaf-to-leaf (original) links of the (original) tree T . But, it is not obvious that the “image” of M in the current tree T' is a matching of the leaf-to-leaf links of T' .

The image of M w.r.t. T' is $\{u'w' : uw \in M, u' \neq w'\}$. We abuse the notation and use M to denote both M and its image w.r.t. T' , and by an M -link of T' we mean the image of an original M -link w.r.t. T' . Whenever we mention semiclosed trees w.r.t. M , we assume that no simple contractions (see Section 6.6.3) are applicable in the current tree T' . Then, Lemma 6.6.3(1) implies that M is a set of leaf-to-leaf links w.r.t. the current tree T' . Hence, semiclosed trees w.r.t. T' and M are well defined.

Let T'_v be a rooted subtree of T' . We use $M(T'_v)$ to denote the set of M -links of T' that have both ends in T'_v . We use $U(T'_v)$ to denote the set of M -exposed leaves of T'_v , including both original leaf nodes and compound leaf nodes. Let $C(T'_v)$ denote the set of compound non-leaf nodes of T'_v . Moreover, for any vector $x \in \mathbb{R}^E$, we use $\Phi(x, T'_v)$ to denote $\frac{1}{2} \sum_{w \in V(T'_v) \cap \mathcal{R}} x(\delta_E(w))$.² We define the credit of T'_v to be the sum of the credits of the nodes in T'_v plus the sum of the credits of the links in $M(T'_v)$.

²Recall that \mathcal{R} is the set of original non-leaf nodes of T ; thus $V(T'_v) \cap \mathcal{R}$ denotes the set of nodes of T'_v excluding all leaves and all compound nodes.

Observe that the credit of a semiclosed tree T'_v is either $1 + \frac{3}{2}|M(T'_v)| + |U(T'_v)| + |C(T'_v)| + \Phi(y, T'_v)$ (if $r \in V(T'_v) \cap \mathcal{R}$), or $\frac{3}{2}|M(T'_v)| + |U(T'_v)| + |C(T'_v)| + \Phi(y, T'_v)$ (if $r \notin V(T'_v) \cap \mathcal{R}$). We call a semiclosed tree T'_v *good* if its credit is $\geq |\Gamma(M, T'_v)| + 1$.

Lemma 6.6.4 *Let T'_v be a semiclosed tree. If at least one of the following conditions is satisfied, then T'_v is good.*

- $T'_v = T'$
- $C(T'_v) \neq \emptyset$
- $|M(T'_v)| \geq 2$
- $\Phi(y, T'_v) \geq 1$
- $|M(T'_v)| = 1$ and $\Phi(y, T'_v) \geq \frac{1}{2}$.

Proof. First, suppose that the root r is not in $V(T'_v) \cap \mathcal{R}$. This implies that if $T'_v = T'$, then r must be contained in some compound non-leaf node, hence, $C(T'_v) \neq \emptyset$. Then, it can be seen that the difference between the credit of T'_v and $|\Gamma(M, T'_v)| + 1$ is

$$\begin{aligned}
&= \frac{3}{2}|M(T'_v)| + |U(T'_v)| + |C(T'_v)| + \Phi(y, T'_v) - |\Gamma(M, T'_v)| - 1 \\
&= \frac{3}{2}|M(T'_v)| + |U(T'_v)| + |C(T'_v)| + \Phi(y, T'_v) - |M(T'_v)| - |U(T'_v)| - 1 \\
&= \frac{1}{2}|M(T'_v)| + |C(T'_v)| + \Phi(y, T'_v) - 1.
\end{aligned}$$

The above quantity is ≥ 0 if any one of the conditions listed in the lemma is satisfied, hence, the result holds.

If the root r is in $V(T'_v) \cap \mathcal{R}$, then T'_v has one more unit of credit, and again it can be seen that the result holds. \square

Consider a minimally semiclosed tree T'_v and suppose that it is good. Then, by Lemma 6.6.2, $\Gamma(M, T'_v)$ is a cover of T'_v , and moreover, T'_v has enough credit to pay for the contraction of $\Gamma(M, T'_v)$. Thus, the algorithm makes progress whenever there exists a minimally semiclosed tree that is good.

6.6.5 Algorithm in summary

We give a summary of the algorithm in pseudocode. The critical step of the algorithm is to find a good semiclosed tree and a cover of it of appropriate size in polynomial time. The details of this step are presented in Section 6.7. There, we show that if a semiclosed tree T'_w is *not* good, then T'_w and its incident links form a subgraph that we call a deficient 3-leaf tree (this is the key result of this chapter). The algorithm finds all occurrences of deficient 3-leaf trees in polynomial time, and then computes another matching of the leaf-to-leaf links that we denote by M^{new} . Then the algorithm finds a minimally semiclosed tree T'_v w.r.t. M^{new} . We prove that T'_v is good (it has enough credits to pay for the contraction of a cover of size $|\Gamma(M, T'_v)|$) and, moreover, $\Gamma(M^{new}, T'_v)$ is a cover of T'_v of appropriate size.

The algorithm starts with $F := \emptyset$ (F is the set of links picked by the algorithm) and $T' := T$ (T' is the current tree T/F).

Algorithm 6.1: Find an approximately optimal solution for TAP.

```

1 while  $T'$  is not a single node do
2   repeatedly apply simple contractions until no simple contractions are applicable;
3   find a good semiclosed tree  $T'_v$  with a cover  $J$  of size  $|\Gamma(M, T'_v)|$  (Algorithm 6.2 in
   Section 6.7 gives the details for finding such a good semiclosed tree);
4   add  $J$  to  $F$ , contract  $T'_v$  to a new compound node, update  $T'$ ;
5 end

```

6.6.6 Worked example

The working of the algorithm is illustrated in Figure 6.5. Observe that M consists of the two links u_0u_1, u_3u_4 . Moreover, note that no simple contractions apply at the start, and the subtree rooted at v_1 , $T'_{v_1} = T_{v_1}$, is a good semiclosed tree. In the first iteration (of the while loop), T_{v_1} is contracted into the compound node a by adding the M -link u_0u_1 and the link v_0v_1 (shadow of v_0u_0) to F ; thus, a corresponds to T'_{v_1} . Consider the credits for this iteration. The M -link u_0u_1 has $\frac{3}{2}$ credit and the M -exposed leaf v_0 has 1 credit. It can be seen that $\Phi(y, T'_{v_1}) \geq \frac{1}{2}y(\delta_E(v_1)) \geq \frac{1}{2}$, and this gives $\frac{1}{2}$ credit. (The formal algorithm does not refer to the “fractional credits” $\Phi(\cdot, \cdot)$, but our analysis relies on these; since we have not presented the formal algorithm in full, we refer to $\Phi(\cdot, \cdot)$ to justify the working of the algorithm.) Hence, we have ≥ 3 credits, and this pays for contracting the two links and for assigning 1 credit to the compound node a . Next, the subtree rooted at v_2 (that has two leaves a, u_2) is contracted into the compound node b via a simple contraction applied to the link au_2 ; thus, b corresponds to the subtree rooted at v_2 . Consider the

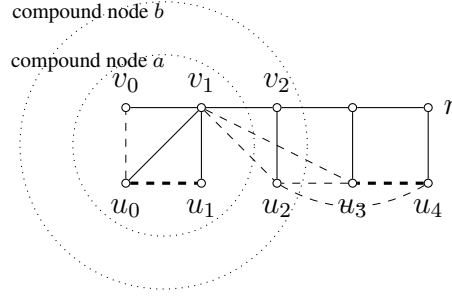


Figure 6.5: Worked example of the algorithm. The edges of the tree T rooted at r are indicated by solid lines, and the maximal links in E are indicated by dashed lines. The matching M is indicated by the thick dashed lines.

credits for this simple contraction. Each of a and u_2 has 1 credit, and these 2 credits pay for contracting one link and for assigning 1 credit to the compound node b . After this, the current tree T' has three leaves, b, u_3, u_4 . Note that T' is a good semiclosed tree. In the final step, T' is contracted into a single compound node by adding to F the links u_3u_4 and br (shadow of bu_4 corresponding to the original link u_2u_4). Consider the credits for the final step. Each of b and r has 1 credit, and the M -link u_3u_4 has $\frac{3}{2}$ credits. Hence, we have $3\frac{1}{2}$ credits, and this pays for contracting the two links and for assigning 1 credit to the resulting compound node. Thus, the algorithm computes the solution $F = \{v_0v_1, u_0u_1, au_2 = v_1u_2, br = u_2r, u_3u_4\}$ of size five; it can be seen that there exists an optimal solution of size four.

6.7 Analysis of the algorithm

This section has our main result. Informally speaking, it asserts the following: if a semiclosed tree T'_v is not good, then T'_v (and its incident links) form a deficient 3-leaf tree.

We assume that we are considering the moment after exhausting simple contractions in the main loop of the algorithm. Thus, Lemma 6.6.3 applies. The analysis mainly consists of two parts. In Section 6.7.1, using local integrality of feasible solutions to the **LAS** system, we show that all semiclosed trees are good, except one particular case that turns out to give deficient 3-leaf trees. Section 6.7.2 shows how to handle the deficient 3-leaf trees. This leads to a polynomial-time algorithm for finding a good semiclosed tree T'_v and a cover of T'_v of size $|\Gamma(M, T'_v)|$.

Deficient 3-leaf tree. Even et al. introduced the notion of “deficient trees”, see [29, Definition 4.7] and Figure 1 of [29]; each of these configurations consists of a rooted subtree with three leaves and some incident links.

Suppose that T'_v is a semiclosed tree with exactly three leaves a, b_1, b_2 . Clearly, among the nodes w of T'_v either there is exactly one node with $\deg_{T'}(w) = 4$ or there are two nodes with degree 3 in T' . In the latter case, we denote these two nodes by u and q ; moreover, we fix the notation such that u is an ancestor of q , and the leaf b_1 is not a descendant of q ; thus, a, b_2 (but not b_1) are descendants of q . In the former case, we denote by u the unique node that is incident to four tree-edges. We call T'_v a *deficient 3-leaf tree* if (i) the link b_1b_2 is present and it is in M , (ii) the link ab_1 is present, and (iii) there exists a link b_2w such that $w \in V(T') - V(T'_v)$.

Moreover, in the first case (with a unique node u in T'_v with $\deg_{T'}(u) = 4$), if conditions (i)–(iii) hold with both labelings (b_1, b_2) and (b_2, b_1) of the M -link, then we fix the notation such that $up(b_2)$ is an ancestor of $up(b_1)$. We call b_2 the *ceiling leaf* of the deficient 3-leaf tree. Hence, for any deficient 3-leaf tree T'_v with ceiling leaf b , it can be seen that $up(b)$ is a proper ancestor of v ; so $up(b)$ is not in T'_v .

We mention that the leaf a may be an original node or a compound node; the properties of the algorithm (see Lemma 6.6.3) ensure that the leaves b_1, b_2 must be original nodes. Note that deficient 3-leaf trees are defined w.r.t. M .

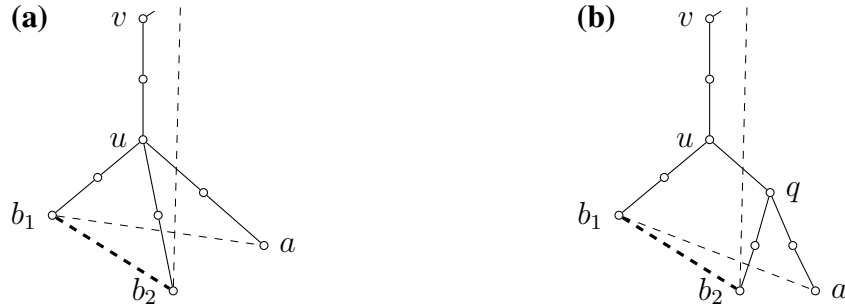


Figure 6.6: Illustration of deficient 3-leaf tree.

6.7.1 Semiclosed trees are good except deficient 3-leaf trees

Let T'_v be a semiclosed tree. We construct an auxiliary graph in order to analyze the credits available in T'_v . We denote the auxiliary graph by $AG(T'_v)$. This is a bipartite graph, and the two sets in the node bipartition are denoted by $ML(T'_v)$ and $AU(T'_v)$. The first set consists of the M -covered leaves of T'_v . The second set contains an auxiliary node \bar{v} (informally speaking, \bar{v} represents the node set $V(T') - V(T'_v)$), as well as all the M -exposed leaves of T'_v , thus, $AU(T'_v) = \{\bar{v}\} \cup U(T'_v)$. We define the edge set of $AG(T'_v)$ as follows: for every link pq (w.r.t.

T'_v) with $p \in \text{ML}(T'_v)$, $q \in U(T'_v)$, the edge pq is in $\text{AG}(T'_v)$, and for every link pq (w.r.t. T'_v) with $p \in \text{ML}(T'_v)$, $q \in V(T'_v) - V(T'_v)$, the edge $p\bar{v}$ is in $\text{AG}(T'_v)$. Thus, $\text{AG}(T'_v)$ is a multigraph, and every edge in $\text{AG}(T'_v)$ corresponds to a link (w.r.t. T'_v). See Figure 6.7 for an example.

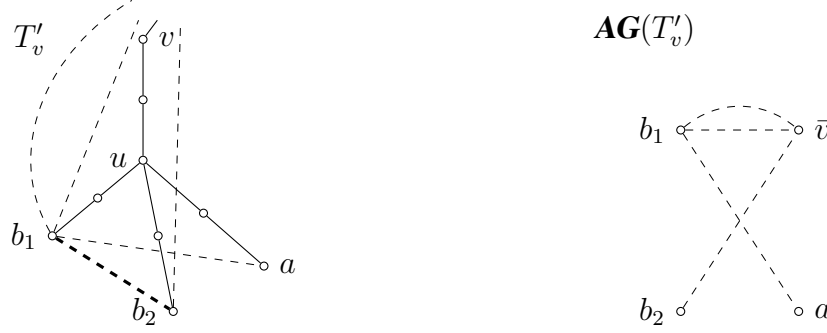


Figure 6.7: Illustration of auxiliary graph. The left figure shows a semiclosed tree T'_v where dashed lines indicate links incident with leaves and the thick dashed line indicates an M -link. The right figure shows the auxiliary graph $\text{AG}(T'_v)$ and its node bipartition $\text{ML}(T'_v) = \{b_1, b_2\}$, $\text{AU}(T'_v) = \{\bar{v}, a\}$.

Lemma 6.7.1 *Suppose that no simple contractions are applicable. Let T'_v be a semiclosed tree such that $T'_v \neq T'$, $C(T'_v) = \emptyset$, and $|M(T'_v)| \leq 1$. Let x be a feasible solution for LP7.*

1. *If $M(T'_v) = \emptyset$, then $\Phi(x, T'_v) \geq 1$. Furthermore, T'_v is good.*
2. *Suppose that $|M(T'_v)| = 1$, and $|U(T'_v)| \geq 1$. Moreover, suppose that x is a feasible solution for LP7 that is integral on $\cup_{w \in \text{ML}(T'_v)} \delta_E(w)$ such that $\Phi(x, T'_v) < \frac{1}{2}$. Then, $|U(T'_v)| = 1$ and the auxiliary graph has a perfect matching such that the corresponding links cover (all the tree-edges in) T'_v , and moreover, $x(\ell) = 1$ for each of the links ℓ in the perfect matching.*

Proof. We start by stating and proving a key claim.

Claim 6.7.2 *Let \bar{J} be a set of links that each have at least one end in T'_v and no end at an M -covered leaf of T'_v ; thus, each link in \bar{J} has at least one end in $V(T'_v) - \text{ML}(T'_v)$. Then, we have $\Phi(x, T'_v) \geq \frac{1}{2}x(\bar{J})$.*

This claim follows from the fact that every link in \bar{J} has an end in $V(T'_v) \cap \mathcal{R}$. To see this, first note that $C(T'_v) = \emptyset$, so every non-leaf node of T'_v is in $V(T'_v) \cap \mathcal{R}$. Consider any link $\ell \in \bar{J}$;

clearly, ℓ has no end at an M -covered leaf of T'_v . If ℓ has an end at a non-leaf node of T'_v , then we are done. Otherwise, ℓ has an end at an M -exposed leaf of T'_v . Since T'_v is semiclosed, ℓ cannot have an end in $V(T') - V(T'_v)$. Moreover, by Lemma 6.6.3(2), no link has both ends at M -exposed leaves. It follows that ℓ has one end in $V(T'_v) \cap \mathcal{R}$. Thus, we proved Claim 6.7.2.

Let \hat{e}_v denote the tree-edge between v and its parent; \hat{e}_v is well defined since $T'_v \neq T'$. Let $J = \delta_E(\hat{e}_v) \cup (\cup_{u \in U(T'_v)} \delta_E(u))$. Then, $x(J) = x(\delta_E(\hat{e}_v)) + \sum_{u \in U(T'_v)} x(\delta_E(u)) \geq 1 + |U(T'_v)|$; the equation holds because (i) T'_v is semiclosed so none of the links in $\delta_E(\hat{e}_v)$ is incident to an M -exposed leaf of T'_v , and (ii) by Lemma 6.6.3(2), no link has both ends at M -exposed leaves; the inequality holds because $x(\delta_E(\hat{e})) \geq 1$ for every tree-edge \hat{e} .

Now, consider the first statement of the lemma. Observe that $M(T'_v) = \emptyset$; also, $U(T'_v) \neq \emptyset$ since T'_v has one or more leaves. By Claim 6.7.2, we have $\Phi(x, T'_v) \geq \frac{1}{2}x(J) \geq \frac{1}{2}(1 + |U(T'_v)|) \geq 1$. Since this inequality holds for every feasible solution x of LP7, it also holds for the feasible solution y of Theorem 6.6.1. Thus, we have $\Phi(y, T'_v) \geq 1$. Hence, by Lemma 6.6.4, T'_v is good.

Finally, consider the second statement of the lemma. Observe that $|M(T'_v)| = 1$ and $|U(T'_v)| \geq 1$. Note that x is integral on $\cup_{w \in ML(T'_v)} \delta_E(w)$, hence, every link in this set has x -value 0 or 1.

Moreover, by Lemma 6.6.3(1), every M -covered leaf is an original node. For every $w \in ML(T'_v)$, since $\delta_E(w)$ is an overlapping clique by Lemma 6.4.3 and x is integral on $\delta_E(w)$, we have $x(\delta_E(w)) \leq 1$. Therefore, $\sum_{w \in ML(T'_v)} x(\delta_E(w)) \leq |ML(T'_v)| = 2$.

Let $\tilde{J} = J - \cup_{w \in ML(T'_v)} \delta_E(w)$. Then, Claim 6.7.2 holds for \tilde{J} in this case, hence, $\Phi(x, T'_v) \geq \frac{1}{2}x(\tilde{J})$.

Note that $x(J) = x(\delta_E(\hat{e}_v)) + \sum_{u \in U(T'_v)} x(\delta_E(u)) \geq 1 + |U(T'_v)|$; this inequality is the same as the inequality used above. Moreover,

$$x(\tilde{J}) \geq x(J) - \sum_{w \in ML(T'_v)} x(\delta_E(w) \cap J) \geq 1 + |U(T'_v)| - |ML(T'_v)| \geq |U(T'_v)| - 1. \quad (6.1)$$

Thus, we have $\Phi(x, T'_v) \geq \frac{1}{2}x(\tilde{J}) \geq \frac{1}{2}(|U(T'_v)| - 1)$.

Clearly, $|U(T'_v)| = 1$, otherwise, we would have $\Phi(x, T'_v) \geq \frac{1}{2}$, thus giving a contradiction. Hence, we have $|ML(T'_v)| = 2 = |AU(T'_v)|$.

Similarly, we claim that each M -covered leaf w of T'_v has a link ℓ_w in $\delta_E(w) \cap J$ of x -value one; otherwise, we would have $\sum_{w \in ML(T'_v)} x(\delta_E(w) \cap J) \leq |ML(T'_v)| - 1$ since every link in $\delta_E(w) \cap J$ takes x -value 0 or 1. This would give the same contradiction (see (6.1) above).

We claim that the set of links $\{\ell_w : w \in ML(T'_v)\}$ maps to the desired perfect matching AM of $AG(T'_v)$. Otherwise, one of the nodes of $AU(T'_v)$ would be incident to two links from

$\{\ell_w : w \in \text{ML}(T'_v)\}$, and we would have $x(J) = x(\delta_E(\hat{e}_v)) + \sum_{u \in U(T'_v)} x(\delta_E(u)) \geq 2 + |U(T'_v)|$, and this would give the same contradiction (see (6.1) above).

Finally, we claim that T'_v is covered by the set of links $\{\ell_w : w \in \text{ML}(T'_v)\}$ that maps to AM . Otherwise, there exists a tree-edge \hat{e} of T'_v that is not covered by this set of links. Let $\delta^+(\hat{e})$ denote the set of links with positive x -value that cover \hat{e} . Clearly, each link in $\delta^+(\hat{e})$ has an end in T'_v , and moreover, has no end in $\text{ML}(T'_v)$; to see the latter assertion, note that each node $w \in \text{ML}(T'_v)$ has $x(\delta_E(w)) \leq 1$ and $x(\ell_w) = 1$, i.e., the nodes in $\text{ML}(T'_v)$ are already “saturated” by the set of links that maps to AM . Thus, Claim 6.7.2 applies to $\delta^+(\hat{e})$ and we have $\Phi(x, T'_v) \geq \frac{1}{2}x(\delta^+(\hat{e})) \geq \frac{1}{2}$, giving the same contradiction. \square

Theorem 6.7.3 *Suppose that no simple contractions are applicable. Let T'_v be a semiclosed tree that is not good. Then T'_v is a deficient 3-leaf tree.*

Proof. Since T'_v is not good, Lemma 6.6.4 implies that $C(T'_v) = \emptyset, |M(T'_v)| \leq 1$, and $T'_v \neq T'$. Then by Lemma 6.7.1(1), we further have $|M(T'_v)| = 1$. Hence, by Lemma 6.6.4 again, $\Phi(y, T'_v) < \frac{1}{2}$.

Let $J = \cup_{w \in \text{ML}(T'_v)} \delta_E(w)$. Note that any node w in $\text{ML}(T'_v)$ is original, by Lemma 6.6.3(1). For any feasible solution $x \in \mathbb{R}^E$ of LP7, we have $|\text{ones}(x) \cap J| \leq 2$ since $|\text{ML}(T'_v)| = 2$. Hence, by Theorem 6.4.1, and the fact that $t \geq 3$, y can be written as a convex combination $\sum_{i \in Z} \lambda_i x^i$ such that $x^i \in \text{Las}_{proj}^{t-2}(\text{LP7})$ and x^i is integral on J for each $i \in Z$. If each x^i , $i \in Z$, has $\Phi(x^i, T'_v) \geq \frac{1}{2}$, then, since y is a convex combination of the x^i , we have $\Phi(y, T'_v) \geq \frac{1}{2}$. This gives a contradiction. Hence, there exists an $i_0 \in Z$ such that $\Phi(x^{i_0}, T'_v) < \frac{1}{2}$.

Case (1) $|M(T'_v)| = 1, U(T'_v) = \emptyset$. Let $uw \in M(T'_v)$, i.e., uw is an M -link with both ends in T'_v . There exists no compound node in the path between u and w in T' , by Lemma 6.6.3. Then, it can be seen that T'_v contains a stem with uw as a twin link. This is a contradiction, because we have an instance of stemless TAP.

Case (2) $|M(T'_v)| = 1, |U(T'_v)| \geq 1$. By Lemma 6.7.1(2), we know $|U(T'_v)| = 1$. (For convenience, we will label the nodes of T'_v using the same labels as in Figure 6.6 but our arguments do not rely on these particular labels.) We denote the M -exposed leaf by a , and the two M -covered leaves by b_1, b_2 , i.e., $\text{ML}(T'_v) = \{b_1, b_2\}$. Now, our goal is to show that T'_v satisfies all the conditions of a deficient 3-leaf tree.

Since $T'_v \neq T'$, let \hat{e}_v denote the tree-edge between v and its parent. By Lemma 6.7.1(2), there exist two links $\ell_v \in \delta_E(\hat{e}_v)$ and $\ell_a \in \delta_E(a)$ such that $x^{i_0}(\ell_v) = x^{i_0}(\ell_a) = 1$, these

two links cover T'_v , and moreover, each of b_1, b_2 is incident to exactly one of these two links (since the auxiliary graph has a perfect matching formed by these two links).

If there is only one non-leaf node u in T'_v with $\deg_{T'}(u) \neq 2$ (see Figure 6.6(a)), then we are done. Otherwise, we have exactly two non-leaf nodes u, q in T'_v with $\deg_{T'}(u) \neq 2$ and $\deg_{T'}(q) \neq 2$. In fact, we must have $\deg_{T'}(u) = 3 = \deg_{T'}(q)$ since T'_v has exactly 3 leaves. W.l.o.g., we assume that u is an ancestor of q . Then, T'_q has only two leaves. By the argument in Case (1), the M -link in T'_v cannot have its two ends at the two leaves of T'_q . This implies that one leaf of T'_q is M -exposed; thus a is a leaf of T'_q . W.l.o.g., take the other leaf of T'_q to be b_2 . Thus, the third leaf b_1 is not in T'_q .

Suppose that ℓ_v is incident to b_1 and ℓ_a is incident to b_2 . Then, the tree-edge between q and its parent is not covered by these two links (see Figure 6.8(a)). This is a contradiction. Hence, ℓ_v is incident to b_2 and ℓ_a is incident to b_1 (see Figure 6.8(b)). Therefore, T'_v satisfies all the conditions of a deficient 3-leaf tree.

□

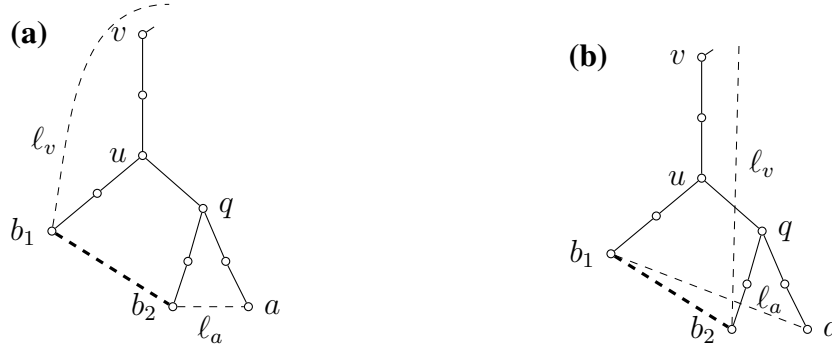


Figure 6.8: The links ℓ_v and ℓ_a in the proof of Theorem 6.7.3.

6.7.2 Addressing deficient 3-leaf trees

Even et al., see [29, Section 4.3], presented an elegant method for addressing deficient 3-leaf trees. We use essentially the same method in this section. The key point is to (temporarily) replace the matching M by another matching of the leaf-to-leaf links denoted by M^{new} .

For a deficient 3-leaf tree T'_w , if T'_w is not a proper subtree of another deficient 3-leaf tree, then we call T'_w a *maximal deficient 3-leaf tree*. By Property 6.2.1, any two different maximal deficient

3-leaf trees are disjoint. To construct M^{new} , we start with $M^{new} := M$, then we examine each maximal deficient 3-leaf tree T'_w and we replace the unique link of $M(T'_w)$ by another leaf-to-leaf link. In more detail, consider any maximal deficient 3-leaf tree T'_w , and let the three leaves be a, b, d , where a is M -exposed, b is the ceiling leaf, and bd is the unique link in $M(T'_w)$; we keep the link ad in M^{new} instead of the M -link bd (see Figure 6.9). Since any two different maximal deficient 3-leaf trees are disjoint, this replacement takes place independently for each maximal deficient 3-leaf tree.

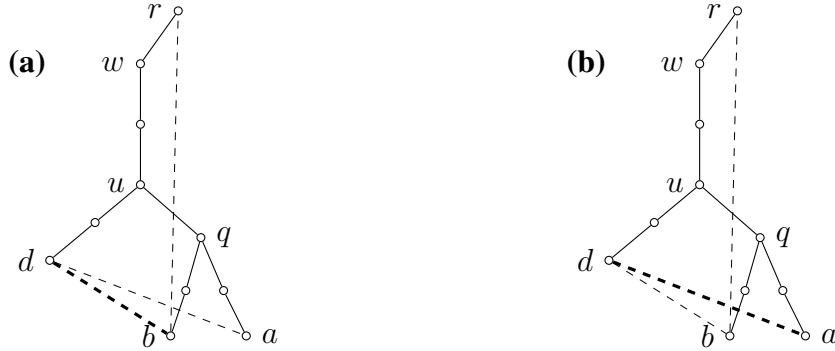


Figure 6.9: Addressing deficient 3-leaf trees by replacing M by M^{new} . Figure (a) shows a maximal deficient 3-leaf tree T'_w with ceiling leaf b and M -exposed leaf a . We obtain the matching M^{new} from M by replacing the link db by the link da ; see Figure (b). T'_w is not a semiclosed tree w.r.t. M^{new} since b is M^{new} -exposed. Instead, T'_r is a minimally semiclosed tree w.r.t. M^{new} . Note that T'_r is not a deficient 3-leaf tree.

Theorem 6.7.4 *Suppose that no simple contractions are applicable. Let T'_v be a minimally semiclosed tree w.r.t. M^{new} . Then T'_v is a good semiclosed tree w.r.t. M and T'_v has a cover $\Gamma(M^{new}, T'_v)$ of size $|\Gamma(M, T'_v)|$.*

Proof. We start by stating and proving a key claim.

Claim 6.7.5 *If T'_v has a node in a maximal deficient 3-leaf tree, then T'_v properly contains this maximal deficient 3-leaf tree.*

Let T'_v share a node with a maximal deficient 3-leaf tree T'_w ; let b denote its ceiling leaf. Note that $up(b)$ is not in T'_w . Observe that b is an M^{new} -exposed node. By the definition of deficient 3-leaf tree, no subtree of T'_w is semiclosed w.r.t. M^{new} . This implies that T'_v cannot be a subtree

of T'_w . Since T'_v and T'_w share a node, by Property 6.2.1, T'_v properly contains T'_w . This proves Claim 6.7.5.

Claim 6.7.5 shows that T'_v cannot be a deficient 3-leaf tree. Otherwise, T'_v is contained in some maximal deficient 3-leaf tree. This contradicts the claim.

Note that any two maximal deficient 3-leaf trees are disjoint, and the replacement of M -links takes place locally in each maximal deficient 3-leaf tree. Hence, it can be seen that T'_v is semiclosed w.r.t. M (because T'_v is semiclosed w.r.t. M^{new}); moreover, we have $|\Gamma(M, T'_v)| = |\Gamma(M^{new}, T'_v)|$ because $|M(T'_v)| = |M^{new}(T'_v)|$.

Since T'_v is not a deficient 3-leaf tree, Theorem 6.7.3 implies that T'_v is good.

Since T'_v is a minimally semiclosed tree w.r.t. M^{new} , Lemma 6.6.2 implies that $\Gamma(M^{new}, T'_v)$ is a cover of T'_v . Moreover, the size of $\Gamma(M^{new}, T'_v)$ is equal to $|\Gamma(M, T'_v)|$. This completes the proof. □

The procedure for finding a good semiclosed tree is summarized in the following pseudocode.

Algorithm 6.2: Find a good semiclosed tree by addressing all deficient 3-leaf trees.

```

1 start with  $M^{new} := M$ ;
2 for each maximal deficient 3-leaf tree  $T'_w$  do
3   | let  $b$  be the ceiling leaf,  $a$  be the  $M$ -exposed leaf, and  $db$  be the  $M$ -link in  $T'_w$ ;
4   | update  $M^{new}$  by replacing  $db$  by  $da$  ( $M^{new} := M^{new} - \{db\} \cup \{da\}$ );
5 end
6 find a minimally semiclosed tree  $T'_v$  w.r.t.  $M^{new}$  (note that  $M^{new}$  is a matching of the
   leaf-to-leaf links);
7  $T'_v$  is a good semiclosed tree w.r.t.  $M$  with a cover  $\Gamma(M^{new}, T'_v)$  of size  $|\Gamma(M, T'_v)|$  by
   Theorem 6.7.4;
```

The discussion above shows how to find a good semiclosed tree T'_v with a cover of size $|\Gamma(M, T'_v)|$ in polynomial time, for the main loop in our algorithm in Section 6.6. Therefore, Algorithm 6.1 (the overall algorithm) runs in polynomial time and returns a solution for TAP with size $\leq (\frac{3}{2} + \epsilon)y(E)$. This proves Theorem 6.6.1.

6.8 Tight example for the analysis

In this section, we present a tight example to show that the approximation guarantee of our algorithm cannot be improved beyond $\frac{3}{2}$. The example has a parameter k . It consists of one

initial block and k copies of a repeated block. See Figure 6.10.

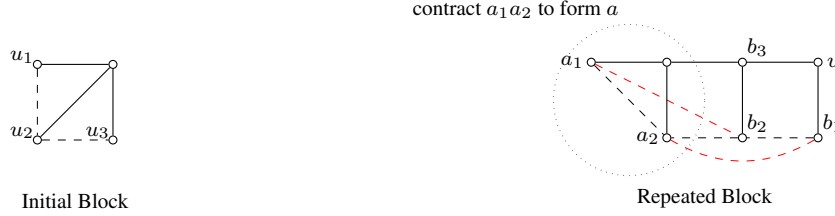


Figure 6.10: Two building blocks of our example. The maximal links are indicated by dashed lines.

Figure 6.11 shows an instance with $k = 3$ copies of the repeated block and all maximal links in E are shown in the figure. Clearly, our instance has no stems. An instance for large k can be constructed by adding more copies of the repeated block. The root node r is always in the rightmost copy of the repeated block.

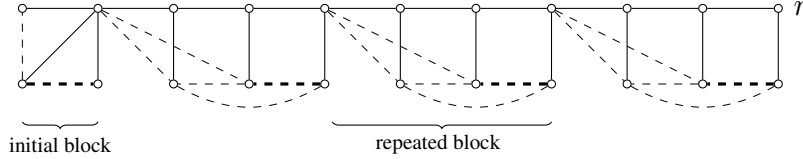


Figure 6.11: Instance for $k = 3$. The tree-edges of T are indicated by solid lines, and the maximal links in E are indicated by dashed lines. The maximum matching M is indicated by thick dashed lines.

In our example, the blocks are disjoint in terms of both tree-edges and links. This implies that any feasible solution of TAP must cover each block separately. To find an optimal solution, we only need to consider each block individually. For the initial block, we need two links to cover the three tree-edges. For the repeated block, we again need two links to cover the six tree-edges (see the links a_1b_2 , a_2b_1 in Figure 6.10). Hence, an optimal solution of our instance has size $2k + 2$.

Now consider the execution of our algorithm on this instance. The maximum matching M is shown in Figure 6.11. At the start, there is no good link available for simple contractions. The initial block is a minimally semiclosed tree (but not a deficient 3-leaf tree). So this block will be contracted via the M-link in it and the link from the M -exposed leaf to the root of the block. After that, we enter into the repeated block (see Figure 6.10). At this moment, the link a_1a_2 connects two M -exposed leaves, which implies it is a good link; note that there are no other

good links. Hence, we apply a simple contraction on the link a_1a_2 to form a compound node a (see Figure 6.10). Then, the repeated block forms a minimally semiclosed tree T'_v with 3 leaves a, b_1, b_2 and one M -link b_2b_1 (see Figure 6.10). Note that T'_v is not a deficient 3-leaf tree since there exists no link between an M -covered leaf of T'_v and a node not in T'_v . Thus T'_v is good. In the next step, T'_v will be contracted via the two links av (shadow of ab_1) and b_1b_2 . After that, we enter into another repeated block, and we apply the same steps as for the previous repeated block. The algorithm applies these iterations till it terminates.

During the running of the algorithm, we use 2 links for the initial block and 3 links for each repeated block (one link for simple contraction and two links for contracting a good semiclosed tree with 3 leaves). Hence, the algorithm returns a solution of size $2 + 3k$. Therefore, the approximation guarantee of the algorithm is $\frac{2+3k}{2+2k}$. When k is sufficiently large, the approximation guarantee approaches $\frac{3}{2}$. This shows that the approximation guarantee of our algorithm cannot be improved beyond $\frac{3}{2}$.

Note that the above instance of TAP has some cut nodes. But we can modify the construction to get a 2-node connected instance by adding some links to the above instance. We add a link from the leaf incident with both maximal links in the initial block (see node u_2 in Figure 6.10) to the non-leaf child of the root of the first repeated block (see node b_3 in Figure 6.10). Moreover, for each pair of consecutive repeated blocks, we add a link between the non-leaf children of their roots. It is not hard to see that the addition of these links does not change the working of the algorithm. Clearly, the addition of these links cannot increase the size of an optimal solution. It follows that the approximation guarantee of the algorithm is at least $\frac{2+3k}{2+2k}$, even for 2-node connected instances.

Proposition 6.8.1 *The instance presented above shows that the algorithm in Section 6.6 cannot provide an approximation guarantee better than $\frac{3}{2}$.*

Chapter 7

Approximating (Unweighted) Tree Augmentation via Lasserre System, Part II

In this chapter,¹ we go deeper into the techniques employed in Chapter 6, and prove the same approximation guarantee of $(\frac{3}{2} + \epsilon)$ for (general) TAP. This chapter follows the same outline as Chapter 6. Moreover, the initial LP and Lasserre tightening are the same for both chapters. Our algorithm in this chapter follows the scheme of the algorithm shown in Chapter 6, a variant of the algorithm of [29] (see Section 6.1, Section 6.6.5); in fact, both our algorithm and the algorithm of [44] follow the same algorithmic scheme of [29] although there are some differences since we are using a solution of the Las system in our analysis; see Section 7.5 for details.

Our algorithm in this chapter is also “combinatorial” and we do *not* need to solve the initial LP nor its Las tightening to run the algorithm. But our analysis relies on the Las system and the decomposition theorem of Karlin et al. [41]. In fact, we do not know how to prove an approximation guarantee of $(\frac{3}{2} + \epsilon)$ for TAP based on a mathematical programming relaxation *without* using the decomposition theorem (Theorem 6.4.1). Our critical use of the decomposition theorem is in proving the assertion “semiclosed trees without sufficient credit implies presence of deficient tree”, see Theorems 6.7.3 and 7.6.6. In Chapter 6, we prove Theorem 6.7.3 by applying the Theorem 6.4.1 to decompose a fractional solution y of the Las system into feasible solutions that are integral over a particular set of links J (local integrality). Note that the size of J need not be $O(1)$, and it is possible that $|J| = \Omega(|V|)$. Our proof of Theorem 6.7.3 makes essential use of the local integrality property on link sets of “unrestricted” size; of course, the key to our approach is to show that $\text{ones}(x) \cap J$ has size $O(1)$ for every feasible solution x of our initial LP. This chapter uses the decomposition theorem in a similar way. Other Lift-and-Project systems

¹The results in this chapter are based on a joint-authored paper [13].

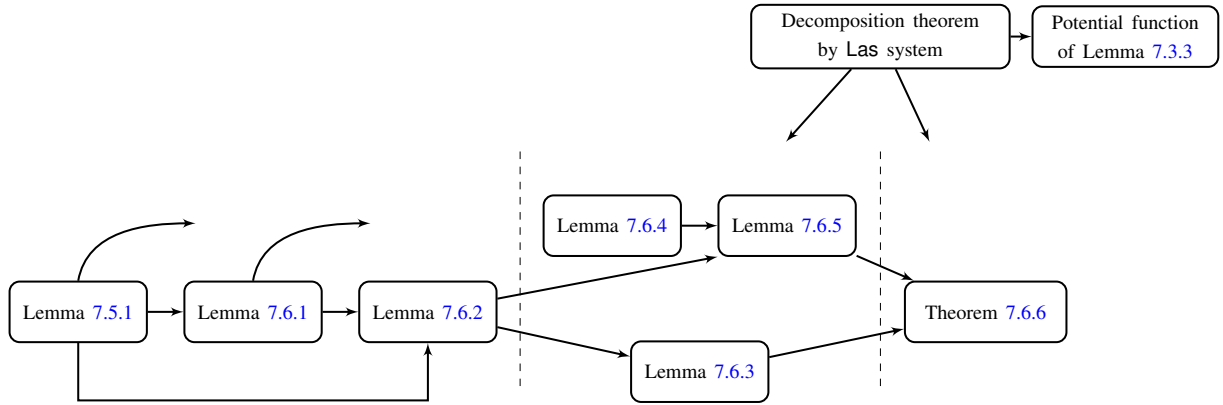


Figure 7.1: Illustration of the argument flow related to the decomposition theorem by Las system. The argument consists of three blocks: low-level, middle-level, and high-level results shown from left to right in the figure.

shown in Section 2.2 are weaker than the Las system, and, to the best of our knowledge, the local integrality property used in our proof of Theorem 6.7.3 does not hold for $O(1)$ levels of any other Lift-and-Project system in Section 2.2.

We also use the Las system and the decomposition theorem to derive our potential functions, see Section 6.5 and Section 7.3, but this use of the decomposition theorem can be “bypassed” because our potential functions may be derived using weaker Lift-and-Project systems such as the LS system or the SA system. Another way for deriving the potential function in Section 6.5 is based on formulating a stronger LP relaxation by adding constraints such as the “matching polytope” constraints on the leaf-to-leaf links.

An outline of this chapter is as follows. Section 7.1 has definitions and notation that are new to this chapter (and not needed in Chapter 6). Section 7.2 discusses the Las tightening of the initial LP, and proves some new properties and inequalities. Section 7.3 derives our potential function, based on a solution of the Las tightening; this potential function differs significantly from the potential function of Chapter 6. Section 7.4 starts the presentation of the algorithm (and credits) by elaborating on two preprocessing steps. Section 7.5 completes the discussion of the algorithm (and credits) by presenting the main loop of the algorithm. The most important component of this chapter is Section 7.6; this section presents the analysis of the algorithm by first proving some low-level properties, then builds on this to prove some intermediate-level lemmas, and then proves the key theorem on deficient trees (Theorem 7.6.6); this section also presents and proves the last piece of the algorithm, namely the handling of deficient trees. Our analysis in Section 7.6 makes essential use of the Las system and the decomposition theorem, see Figure 7.1.

7.1 Preliminaries and notation

We follow the definitions and notation given in Section 6.2 except the notation \mathcal{R} and the definitions of “stem” and “twin links”. Additionally, in this section, we present some definitions and notation that are new to this chapter.

By the argument in Section 6.2, we can make the following assumption.

Assumption: E is shadow-closed.

Recall that for any $U \subseteq V$, we denote the set of links with both ends in U by $E(U)$, and for any two subsets U, W of V , we denote the set of links with one end in U and the other end in W by $E(U, W)$; thus, $E(U, W) := \{uw \in E : u \in U, w \in W\}$. We use similar notation for some subsets of E ; for example, E^{reg} denotes a particular subset of E (defined below), and $E^{reg}(U, W)$ denotes $\{uw \in E^{reg} : u \in U, w \in W\}$.

One of the nodes r of T is designated as the root; thus, we have a rooted tree (T, r) . Throughout this chapter, we use L to denote the set of leaves of T . Let $L(T_v)$ be the set of the leaves in T_v . The terms *tree* or *subtree* refer to a rooted subtree.

Stems and twin links

We call a node s of T a *stem* if s is not the root r , s has exactly two children, s has exactly two descendants that are leaves, and there exists a link in E between the two leaves of T_s ; we call the link between the two leaves of T_s a *twin link*, and denote it by $twinkl(s)$; this differs from Chapter 6 (see Remark 6.2.2 at the end of Section 6.2). Let E^{twin} denote the set of twin links. Observe that there is a one-to-one correspondence between twin links and stems. (The notion of stems and twin links is due to [29].)

Throughout, we use \mathcal{S} to denote the set of stems; thus, $\mathcal{S} = \{v \in V : v \text{ is a stem of } T\}$. Moreover, we use \mathcal{R} to denote the set of nodes that are neither stems nor leaves; thus $\mathcal{R} = V - (\mathcal{S} \cup L)$. The definition of \mathcal{R} in this chapter differs from Chapter 6; this is because we may have stems in the input instance of TAP in this chapter.

For any stem node s , we define $\delta_E^{out}(s) = \{vs \in E : v \notin V(T_s)\}$. Similarly, we define $\delta_E^{in}(s) = \{vs \in E : v \in V(T_s)\}$.

Buds and buddy links

Besides stems and their associated subtrees, one other type of node plays an important role in this chapter.

We call a leaf b_0 a *bud* (see Figure 7.2) if there exists a (rooted) subtree T_v with exactly three leaves b_0, b_1, b_2 such that (i) $up(b_0)$ is a descendant of v (possibly, $up(b_0) = v$), (ii) $T_{up(b_0)}$ contains a stem s such that b_0, b_1 are the leaves of T_s , and (iii) the link b_1b_2 exists. We call b_1b_2 the *buddy link* of b_0 and denote it by $buddy\text{lk}(b_0)$. Observe that there exists an ancestor q of s in T_v such that q is the least common ancestor of s and b_2 ; possibly, $q = v$, and possibly, $q = v = r$; moreover, $L(T_{up(b_0)}) = \{b_0, b_1\}$ or $L(T_{up(b_0)}) = \{b_0, b_1, b_2\}$. Note that b_1 may be a bud as well. In that case, each leaf of $T_{up(b_1)}$ is in $\{b_0, b_1, b_2\}$, there exists a link between b_0, b_2 , and $buddy\text{lk}(b_1) = b_0b_2$.

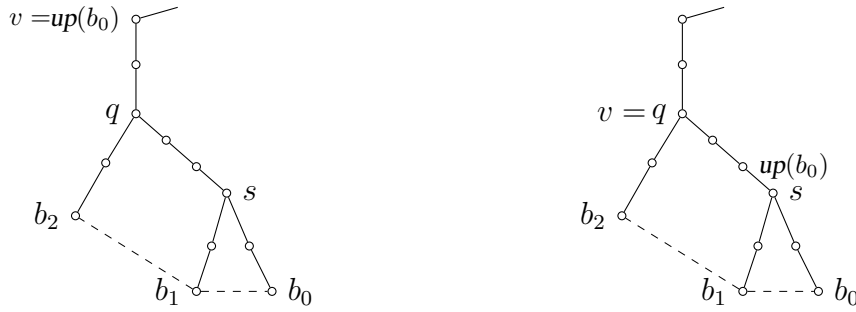


Figure 7.2: Illustration of a bud b_0 in the subtree T_v . The figure on the left-hand side shows the case when $up(b_0)$ is an ancestor of all three leaves in T_v . The figure on the right-hand side shows the other case.

Consider a bud b_0 , and let s, q be as above; we define

$$\mathcal{R}^{special}(b_0) := V(P_{b_0, up(b_0)}) - V(P_{s, q}) - \{b_0\}.$$

Thus $\mathcal{R}^{special}(b_0)$ consists of the internal nodes on the tree-path between b_0 and s , and all nodes on the tree-path between the parent of q and $up(b_0)$ if $up(b_0)$ is a proper ancestor of q (see Figure 7.3).

Fact 7.1.1 $\mathcal{R}^{special}(b_0)$ is the set of nodes w such that there exists a link b_0w in E , and w is not on the tree-path of the link $buddy\text{lk}(b_0)$. Every node in $\mathcal{R}^{special}(b_0)$ is an ancestor of b_0 and also has a unique child. For two buds b_0, \bar{b}_0 associated with two distinct stems respectively, $\mathcal{R}^{special}(b_0) \cap \mathcal{R}^{special}(\bar{b}_0) = \emptyset$.

We denote the set of buds by L^{bud} . We denote the set of buddy links by E^{buddy} . Observe that there is a unique buddy link for each bud; thus, there is a bijection between L^{bud} and E^{buddy} . For

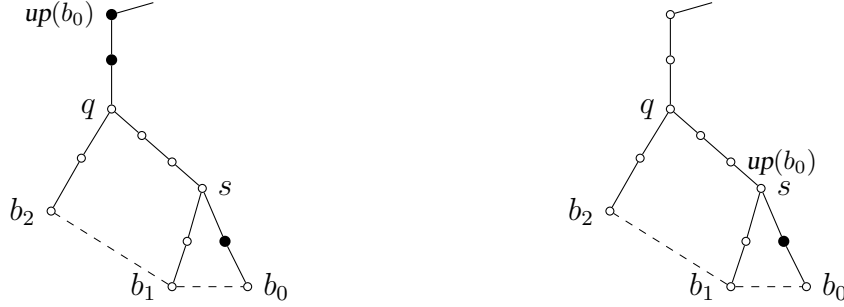


Figure 7.3: Illustration of $\mathcal{R}^{special}(b_0)$ for a bud b_0 . The nodes in $\mathcal{R}^{special}(b_0)$ are indicated by the dark solid ones. The figure on the left-hand side shows the case when $up(b_0)$ is an ancestor of all associated three leaves. The figure on the right-hand side shows the other case.

any node w of the tree T , we denote by $L^{bud}(w)$ the set of buds in the tree rooted at w , T_w . If s is a stem, then note that $L^{bud}(s)$ may contain zero, one, or two nodes.

Let $\mathcal{R}^{special} = \cup_{b \in L^{bud}} \mathcal{R}^{special}(b)$ and $\mathcal{R}^{nonspl} = \mathcal{R} - \mathcal{R}^{special}$. Thus, we partition the set \mathcal{R} (of nodes that are neither stems nor leaves) into two subsets, the “special” subset $\mathcal{R}^{special}$ and the “normal” subset \mathcal{R}^{nonspl} .

We use E^{reg} to denote the set $E - (E^{twin} \cup E^{buddy})$, namely, the set of links that are neither twin links nor buddy links.

7.2 Lasserre tightening and its properties

We apply the **LAS** system to the same initial LP (**LP7**) in Section 6.3. Section 6.4 contains some basic properties and inequalities on the **LAS** tightening of **LP7**. In this section, we present two new results.

Lemma 7.2.1 *Let x be a feasible solution of **LP7** and let s be a stem. Then $|\text{ones}(x) \cap \delta_E(s)| \leq 3$.*

Proof. Since s is a stem, s is incident to three tree-edges. Let $\hat{e}_1, \hat{e}_2, \hat{e}_3$ be the tree-edges incident with s . Consider $\delta_E(\hat{e}_i)$ for $1 \leq i \leq 3$. Clearly, $J_i = \delta_E(\hat{e}_i) \cap \delta_E(s)$ is an overlapping clique and $\delta_E(s) = \cup_{1 \leq i \leq 3} J_i$. Then the result follows from Lemma 6.4.2. \square

Lemma 7.2.2 *Let $t \geq 3$, and let $y \in \text{Las}_{proj}^t(\text{LP7})$ be the projection of a feasible solution of the level t of the **LAS** system. Suppose that s is a stem with two leaves b_0, b_1 in T_s .*

1. For the twin link $\text{twinkl}(s) = b_0b_1$ of s , we have $y(b_0b_1) \leq y(\delta_E^{\text{out}}(s))$.
2. For a bud b in T_s (if exists), we have $y(\text{buddylk}(b)) \leq y(bs) + y(E(b, \mathcal{R}^{\text{special}}(b)))$.

Proof. Let $J = \delta_E(b_0) \cup \delta_E(b_1)$. Since b_0, b_1 are both leaves, by Lemma 6.4.3, $\delta_E(b_0)$ and $\delta_E(b_1)$ are overlapping cliques. Thus, by Lemma 6.4.2, $|\text{ones}(x) \cap J| \leq 2$ for any feasible solution x of LP7. By Theorem 6.4.1, y can be written as a convex combination $\sum_{i \in Z} \lambda_i x^i$ such that $x^i \in \text{Las}_{\text{proj}}^1(\text{LP7})$ and $x^i|_J$ is integral, $\forall i \in Z$.

(1) Since $b_0b_1 \in J$, we have either $x^i(b_0b_1) = 0$ or $x^i(b_0b_1) = 1$. Let $Z_1 = \{i : x^i(b_0b_1) = 1\}$. Then $y(b_0b_1) = \sum_{i \in Z_1} \lambda_i x^i(b_0b_1)$.

Consider x^i for $i \in Z_1$. Let \hat{e}_s the tree-edge between s and its parent. Then, $x^i(\delta_E(\hat{e}_s)) \geq 1$. Notice that every link in $\delta_E(\hat{e}_s)$ with positive x^i -value must have s as its end in T_s ; otherwise, if such a link has an end at some other node of T_s , it will be overlapping with the link b_0b_1 , thus contradicting the constraints on overlapping cliques in LP7. Thus, $x^i(\delta_E(\hat{e}_s)) = x^i(\delta_E(\hat{e}_s) \cap \delta_E(s))$. Hence, $x^i(\delta_E^{\text{out}}(s)) = x^i(\delta_E(\hat{e}_s) \cap \delta_E(s)) = x^i(\delta_E(\hat{e}_s)) \geq 1 = x^i(b_0b_1), \forall i \in Z_1$. Consequently, $y(b_0b_1) = \sum_{i \in Z_1} \lambda_i x^i(b_0b_1) \leq \sum_{i \in Z_1} \lambda_i x^i(\delta_E^{\text{out}}(s)) \leq \sum_{i \in Z} \lambda_i x^i(\delta_E^{\text{out}}(s)) = y(\delta_E^{\text{out}}(s))$.

(2) Without loss of generality, suppose b_0 is the bud b . Let b_1b_2 be the buddy link $\text{buddylk}(b_0)$. Since $b_1b_2 \in J$, we have either $x^i(b_1b_2) = 0$ or $x^i(b_1b_2) = 1$. Let $Z_1^b = \{i : x^i(b_1b_2) = 1\}$. Then $y(b_1b_2) = \sum_{i \in Z_1^b} \lambda_i x^i(b_1b_2)$.

Consider x^i for $i \in Z_1^b$. Clearly, there is a link $\ell = b_0w$ incident with b_0 such that $x^i(\ell) = 1$. If w is in P_{b_1, b_2} but not s , then ℓ is overlapping with b_1b_2 , which contradicts the constraints on overlapping cliques in LP7. Hence, by Fact 7.1.1, $w \in \mathcal{R}^{\text{special}}(b_0) \cup \{s\}$. Then, $x^i(\text{buddylk}(b_0)) = 1 \leq x^i(\ell) \leq x^i(b_0s) + x^i(E(b_0, \mathcal{R}^{\text{special}}(b_0)))$.

Thus, $y(\text{buddylk}(b_0)) = \sum_{i \in Z_1^b} \lambda_i x^i(b_1b_2) \leq \sum_{i \in Z_1^b} \lambda_i (x^i(b_0s) + x^i(E(b_0, \mathcal{R}^{\text{special}}(b_0)))) \leq \sum_{i \in Z} \lambda_i (x^i(b_0s) + x^i(E(b_0, \mathcal{R}^{\text{special}}(b_0)))) = y(b_0s) + y(E(b_0, \mathcal{R}^{\text{special}}(b_0)))$. \square

7.3 Potential function

This section presents the potential function used by the analysis of our algorithm; the potential function is based on a feasible solution y to the Las tightening of LP7. The potential function does not apply to feasible solutions of LP7. Thus, it is essential for our results/analysis to tighten LP7. Possibly, our potential function can be obtained by applying weaker Lift-and-Project systems (e.g., Lovász-Schrijver, or Sherali-Adams) to LP7. But this does not suffice for our

analysis in Sections 7.5–7.6, because our analysis makes essential use of the decomposition theorem (Theorem 6.4.1), and that result is not known to hold for other weaker Lift-and-Project systems.

Our potential function is defined via a subset of the leaves that is denoted by Λ . This subset is determined by the instance of TAP (informally speaking, it consists of all the leaves of all occurrences of a particular type of subtree, called a bad 2-stem tree ... see Section 7.4). Thus, our potential function consists of two parts, a “preprocessing” part and a “normal” part. Our algorithm applies two preprocessing steps, and one of the preprocessing steps contracts all occurrences of the maximal bad 2-stem trees, and for this we have to “charge” the “preprocessing” part of our potential function.

Let Λ be a set of leaves. We say that Λ is *compatible* if it satisfies the following:

- For every twin link and for every buddy link, either both ends of the link are in Λ or none of the ends of the link are in Λ ; in other words, no twin link and no buddy link is present in $E(\Lambda, L - \Lambda)$.

In what follows, let Λ denote a compatible subset of L .

We denote the set of stems with both leaves in Λ by \mathcal{S}_Λ . Similarly, let L_Λ^{bud} denote the set of buds in Λ , and let $L_{(L-\Lambda)}^{bud}$ denote the set of buds in $L - \Lambda$. By the definition of a compatible set, the following fact holds.

Fact 7.3.1

$$\begin{aligned} L_{(L-\Lambda)}^{bud} &= \cup_{s \in \mathcal{S} - \mathcal{S}_\Lambda} L^{bud}(s), \\ E^{twinkl}(L - \Lambda) &= \{twinkl(s), s \in \mathcal{S} - \mathcal{S}_\Lambda\}, \\ E^{buddylk}(L - \Lambda) &= \cup_{s \in \mathcal{S} - \mathcal{S}_\Lambda} \{buddylk(b) : b \in L^{bud}(s)\}. \end{aligned}$$

Let $\widehat{M}_{(L-\Lambda)}^{reg}$ denote a maximum matching of the subgraph $(L - \Lambda, E^{reg}(L - \Lambda))$, and let $\widehat{U}_{(L-\Lambda)}$ denote the set of nodes of this subgraph exposed by the matching $\widehat{M}_{(L-\Lambda)}^{reg}$; thus $\widehat{U}_{(L-\Lambda)} = (L - \Lambda) - \{v \in V : v \text{ is an end of some link } \ell \in \widehat{M}_{(L-\Lambda)}^{reg}\}$. We mention that our potential function (the right-hand side of the inequality in Lemma 7.3.3 below) refers to the terms Λ , $\widehat{M}_{(L-\Lambda)}^{reg}$, $\widehat{U}_{(L-\Lambda)}$. Thus, when we use our potential function, we have to ensure that these terms have been defined already; we will “fix” our potential function by appropriately defining Λ , $\widehat{M}_{(L-\Lambda)}^{reg}$, $\widehat{U}_{(L-\Lambda)}$ in Section 7.4.

Given Λ and $y \in \mathbb{R}^E$, we use $lbd_y(\Lambda)$ to denote the quantity

$$\frac{3}{2}y(E(\Lambda)) + \frac{1}{2}y(E(\Lambda, L-\Lambda)) + y(E(\Lambda, V-L)) + \frac{1}{2} \sum_{s \in \mathcal{S}_\Lambda} y(\delta_E(s));$$

this is one of the terms in our potential function; informally speaking, this is the main component of the “preprocessing” part of the potential function and it is associated with the set Λ .

For any stem s and for any $x \in \mathbb{R}^E$, we define $slack_x(s)$ to be

$$\frac{1}{2} \left(x(\delta_E^{in}(s)) + \sum_{b \in L^{bud}(s)} (x(E(b, \mathcal{R}^{special}(b))) - x(buddylk(b))) \right) + \frac{1}{2} \left(x(\delta_E^{out}(s)) - x(twinkl(s)) \right).$$

By Lemma 7.2.2, the following fact holds.

Fact 7.3.2 *For any stem s and for any feasible solution y to $\text{Las}_{proj}^t(\text{LP7})$ where $t \geq 3$, we have $slack_y(s) \geq 0$.*

Lemma 7.3.3 *Let $\epsilon > 0$ be a constant, and let $t \geq \max\{\frac{1}{2\epsilon} + 1, 3\}$. Let $y \in \text{Las}_{proj}^t(\text{LP7})$. Then,*

$$\begin{aligned} \left(\frac{3}{2} + \epsilon\right)y(E) &\geq \frac{3}{2}|\widehat{M}_{(L-\Lambda)}^{reg}| + |\widehat{U}_{(L-\Lambda)}| + lbd_y(\Lambda) + \frac{1}{2}y(E(V-L)) \\ &\quad + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(E(v, V-L_{(L-\Lambda)}^{bud}(v))) + \sum_{s \in \mathcal{S}-\mathcal{S}_\Lambda} slack_y(s), \end{aligned}$$

where $slack_y(s) \geq 0$ for $s \in \mathcal{S}$.

Proof. By Fact 7.3.2, we have $slack_y(s) \geq 0$ for $s \in \mathcal{S}$. For each link uw , we distribute the value $\frac{3}{2}y_{uw}$ as follows.

- both $u, w \in L$: then uw keeps the value $\frac{3}{2}y_{uw}$;
- both $u, w \in V-L$: then uw keeps the value $\frac{1}{2}y_{uw}$ and each of the ends u and w gets value $\frac{1}{2}y_{uw}$;
- only one of u or w is in $V-L$: then uw keeps the value y_{uw} and the end in $V-L$ gets value $\frac{1}{2}y_{uw}$.

(In other words, each node in $V-L$ borrows value $\frac{1}{2}y_{uw}$ from each link uw incident to it, and the link keeps the remaining value; links uw that are not incident to $V-L$ keep all of the value $\frac{3}{2}y_{uw}$.)

Thus, we have

$$\begin{aligned} \frac{3}{2}y(E) &= \frac{3}{2}y(E(\Lambda)) + \frac{3}{2}y(E(L-\Lambda)) + \frac{3}{2}y(E(\Lambda, L-\Lambda)) \\ &\quad + \frac{1}{2}y(E(V-L)) + y(E(V-L, \Lambda)) + y(E(V-L, L-\Lambda)) + \frac{1}{2} \sum_{v \in V-L} y(\delta_E(v)). \end{aligned}$$

Then we increase the coefficients of the twin links and buddy links in $E(L-\Lambda)$ from $\frac{3}{2}$ to 2 by borrowing the value $\frac{1}{2}y(E^{twin}(L-\Lambda)) + \frac{1}{2}y(E^{buddy}(L-\Lambda))$ from the last term above, and adding it to the term $\frac{3}{2}y(E(L-\Lambda))$. Thus, we replace the term $\frac{3}{2}y(E(L-\Lambda))$ by $2y(E(L-\Lambda)) - \frac{1}{2}y(E^{reg}(L-\Lambda))$, and we replace the last term by

$$\begin{aligned} &\frac{1}{2} \sum_{v \in V-L} y(\delta_E(v)) - \frac{1}{2}y(E^{twin}(L-\Lambda)) - \frac{1}{2}y(E^{buddy}(L-\Lambda)) \\ &= \frac{1}{2} \sum_{s \in S_\Lambda} y(\delta_E(s)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(\delta_E(v)) + \frac{1}{2} \sum_{s \in S-S_\Lambda} (y(\delta_E^{in}(s)) + y(\delta_E^{out}(s))) \\ &\quad - \frac{1}{2}y(E^{twin}(L-\Lambda)) - \frac{1}{2}y(E^{buddy}(L-\Lambda)) \\ &= \frac{1}{2} \sum_{s \in S_\Lambda} y(\delta_E(s)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(\delta_E(v)) + \frac{1}{2} \sum_{s \in S-S_\Lambda} y(\delta_E^{out}(s)) - \frac{1}{2}y(E^{twin}(L-\Lambda)) \\ &\quad + \frac{1}{2} \sum_{s \in S-S_\Lambda} y(\delta_E^{in}(s)) - \frac{1}{2}y(E^{buddy}(L-\Lambda)) \\ &\stackrel{(1)}{=} \frac{1}{2} \sum_{s \in S_\Lambda} y(\delta_E(s)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \left(\frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(E(v, V-L_{(L-\Lambda)}^{bud}(v))) + \sum_{b \in L_{(L-\Lambda)}^{bud}} y(E(b, \mathcal{R}^{special}(b))) \right) \\ &\quad + \frac{1}{2} \sum_{s \in S-S_\Lambda} y(\delta_E^{out}(s)) - \frac{1}{2}y(E^{twin}(L-\Lambda)) + \frac{1}{2} \sum_{s \in S-S_\Lambda} y(\delta_E^{in}(s)) - \frac{1}{2}y(E^{buddy}(L-\Lambda)) \\ &\stackrel{(2)}{=} \frac{1}{2} \sum_{s \in S_\Lambda} y(\delta_E(s)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) \\ &\quad + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(E(v, V-L_{(L-\Lambda)}^{bud}(v))) + \frac{1}{2} \sum_{s \in S-S_\Lambda} \left(y(\delta_E^{in}(s)) + \sum_{b \in L^{bud}(s)} (y(E(b, \mathcal{R}^{special}(b))) - y(buddylk(b))) \right) \\ &\quad + \frac{1}{2} \sum_{s \in S-S_\Lambda} \left(y(\delta_E^{out}(s)) - y(twinlk(s)) \right) \\ &= \frac{1}{2} \sum_{s \in S_\Lambda} y(\delta_E(s)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(E(v, V-L_{(L-\Lambda)}^{bud}(v))) + \sum_{s \in S-S_\Lambda} slack_y(s). \end{aligned}$$

where (2) follows from Fact 7.3.1 due to the assumption that Λ is compatible and (1) follows from the following equation:

$$\bigcup_{w \in \mathcal{R}^{special}} E(w, L_{(L-\Lambda)}^{bud}(w)) = \bigcup_{b \in L_{(L-\Lambda)}^{bud}} E(b, \mathcal{R}^{special}(b)).$$

This equation is based on some observations. First, the set on the right-hand side is clearly a subset of the set on the left-hand side. Conversely, we consider a link wa in the set on the left-hand side, i.e., $w \in \mathcal{R}^{special}$ and $a \in L_{(L-\Lambda)}^{bud}(w)$. Since w is an ancestor of a and w is not on the tree-path of the link $buddylk(a)$, by Fact 7.1.1, we have w is in $\mathcal{R}^{special}(a)$, i.e., $wa \in E(a, \mathcal{R}^{special}(a))$. Hence, the link wa belongs to the set on the right-hand side as well.

Thus, the expression for $\frac{3}{2}y(E)$ can be written as

$$\begin{aligned} &= \frac{3}{2}y(E(\Lambda)) + \frac{3}{2}y(E(\Lambda, L-\Lambda)) + 2y(E(L-\Lambda)) - \frac{1}{2}y(E^{reg}(L-\Lambda)) \\ &\quad + \frac{1}{2}y(E(V-L)) + y(E(V-L, \Lambda)) + y(E(V-L, L-\Lambda)) \\ &\quad + \frac{1}{2} \sum_{s \in \mathcal{S}_\Lambda} y(\delta_E(s)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(E(v, V-L_{(L-\Lambda)}^{bud}(v))) + \sum_{s \in \mathcal{S}-\mathcal{S}_\Lambda} slack_y(s) \\ &= \left(\frac{3}{2}y(E(\Lambda)) + \frac{1}{2}y(E(\Lambda, L-\Lambda)) + y(E(\Lambda, V-L)) + \frac{1}{2} \sum_{s \in \mathcal{S}_\Lambda} y(\delta_E(s)) \right) \\ &\quad + \left(y(E(L-\Lambda, \Lambda)) + 2y(E(L-\Lambda)) + y(E(L-\Lambda, V-L)) \right) \\ &\quad + \frac{1}{2}y(E(V-L)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(E(v, V-L_{(L-\Lambda)}^{bud}(v))) + \sum_{s \in \mathcal{S}-\mathcal{S}_\Lambda} slack_y(s) \\ &\quad - \frac{1}{2}y(E^{reg}(L-\Lambda)) \\ &= lbd_y(\Lambda) + \sum_{v \in L-\Lambda} y(\delta_E(v)) + \frac{1}{2}y(E(V-L)) \\ &\quad + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(E(v, V-L_{(L-\Lambda)}^{bud}(v))) + \sum_{s \in \mathcal{S}-\mathcal{S}_\Lambda} slack_y(s) \\ &\quad - \frac{1}{2}y(E^{reg}(L-\Lambda)). \end{aligned}$$

By Lemma 6.4.3, we have $y(\delta_E(v)) = 1$ for any $v \in L$. Thus, we can replace the term $\sum_{v \in L-\Lambda} y(\delta_E(v))$ (the 2nd term in the displayed equation above) by $|L-\Lambda|$. Moreover, by Lemma 6.4.4, we have $\frac{y|_{E(L)}}{1+\epsilon}$ is in the matching polytope of $(L, E(L))$. Thus, $\frac{y|_{E^{reg}(L-\Lambda)}}{1+\epsilon}$ is in

the matching polytope of $(L-\Lambda, E^{reg}(L-\Lambda))$, which implies $y(E^{reg}(L-\Lambda)) \leq (1+\epsilon)|\widehat{M}_{(L-\Lambda)}^{reg}|$. We derive the inequality (stated in the lemma) by replacing the term $-\frac{1}{2}y(E^{reg}(L-\Lambda))$ (the last term in the displayed equation above) by $-\frac{1}{2}(1+\epsilon)|\widehat{M}_{(L-\Lambda)}^{reg}|$. Now, observe that $|L-\Lambda| - \frac{1}{2}(1+\epsilon)|\widehat{M}_{(L-\Lambda)}^{reg}| = |\widehat{U}_{(L-\Lambda)}| + \frac{3}{2}|\widehat{M}_{(L-\Lambda)}^{reg}| - \frac{1}{2}(\epsilon)|\widehat{M}_{(L-\Lambda)}^{reg}|$, because $\widehat{U}_{(L-\Lambda)} = (L-\Lambda) - \{v \in V : v \text{ is an end of some link } \ell \in \widehat{M}_{(L-\Lambda)}^{reg}\}$. Note that $|\widehat{M}_{(L-\Lambda)}^{reg}| \leq \frac{1}{2}|L| = \frac{1}{2} \sum_{v \in L} y(\delta_E(v)) \leq y(E)$, hence, $-\frac{\epsilon}{2}|\widehat{M}_{(L-\Lambda)}^{reg}| \geq -\epsilon y(E)$. Thus, we get our potential function:

$$\begin{aligned} \left(\frac{3}{2} + \epsilon\right)y(E) \geq & \frac{3}{2}|\widehat{M}_{(L-\Lambda)}^{reg}| + |\widehat{U}_{(L-\Lambda)}| + \text{lb}_y(\Lambda) + \frac{1}{2}y(E(V-L)) \\ & + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl}} y(\delta_E(v)) + \frac{1}{2} \sum_{v \in \mathcal{R}^{special}} y(E(v, V-L_{(L-\Lambda)}^{bud}(v))) + \sum_{s \in \mathcal{S}-\mathcal{S}_\Lambda} \text{slack}_y(s). \end{aligned}$$

□

7.4 Algorithm and credits I: Preprocessing steps

We state our main result for (unweighted) TAP:

Theorem 7.4.1 *Let $\epsilon > 0$ be any (small) constant, and let $t \geq \max\{17, \frac{1}{2\epsilon} + 1\}$. The integrality ratio of $\text{Las}_{proj}^t(\text{LP7})$ is $\leq \frac{3}{2} + \epsilon$. Moreover, there is a polynomial-time algorithm for TAP that finds a feasible solution (a set of links that covers the tree T) of size $\leq (\frac{3}{2} + \epsilon)y(E)$, where y is an optimal solution of $\text{Las}_{proj}^t(\text{LP7})$.*

For the rest of this chapter, we fix $y \in \mathbb{R}^E$ to be an optimal solution in $\text{Las}_{proj}^t(\text{LP7})$, where $t \geq \max\{17, 1 + \frac{1}{2\epsilon}\}$, where $\epsilon > 0$ is any constant. Our goal is to show that our algorithm finds a set of links that covers T of size $\leq (\frac{3}{2} + \epsilon)y(E)$. We achieve this goal using our potential function (this is the right-hand side of the inequality in Lemma 7.3.3); we will “fix” the potential function below by defining the relevant terms (namely, $\Lambda, \widehat{M}_{(L-\Lambda)}^{reg}, \widehat{U}_{(L-\Lambda)}$). Recall from Section 6.6 that the potential function provides “credit” to the algorithm.

Also, recall from Section 6.6 that the combinatorial algorithm is a greedy-type iterative algorithm that makes a leaves-to-root scan over the tree T and (incrementally) constructs a set of links F that covers T . The algorithm starts with $F := \emptyset$, at each step it adds one or more links to F (it never removes links from F), and at termination, it outputs a cover F of T whose size is \leq the potential function.

The algorithm iteratively finds a set of links $F^{iter} \subseteq E - F$ such that the contraction of F^{iter} in the current tree results in a single new compound node; thus, each contraction creates one new compound node.

The algorithm incurs a cost of one unit for every link that it picks, and it incurs a cost of one unit for each new compound node that it creates in the execution. The key to the analysis is to show that for each step, the cost incurred is compensated by a part of the “credit.”

We mention that the nodes or links that get contracted into a compound node are no longer relevant for the algorithm or the analysis. In particular, the credit (if any) of such nodes or links may be used at the step when they get contracted into a compound node, but after that step, any remaining credit of such nodes or links is not used at all.

7.4.1 Semiclosed trees

We recall the notion of a semiclosed tree w.r.t. an arbitrary matching from Section 6.6.1

Let T'_v be a rooted subtree of the current tree $T' = T/F$. Let \bar{M} be an arbitrary matching of the leaf-to-leaf links. T'_v is called *semiclosed* w.r.t. \bar{M} if the following conditions hold:

- (i) Each link in \bar{M} either has both ends in T'_v or has no end in T'_v .
- (ii) Every link incident to an \bar{M} -exposed leaf of T'_v has both ends in T'_v .

Let $\bar{M}(T'_v)$ denote the set of links in \bar{M} that have both ends in T'_v .

We define

$$\Gamma(\bar{M}, T'_v) := \bar{M}(T'_v) \cup \{up(w)w : w \text{ is an } \bar{M}\text{-exposed leaf in } T'_v\};$$

thus, we associate a “basic link set” with the pair \bar{M}, T'_v . In general, the “basic link set” may not be a cover of T'_v .

By a *minimally semiclosed tree* T'_v we mean that T'_v is semiclosed but none of the proper rooted subtrees of T'_v is semiclosed.

For a rooted tree T'_v and a set of links J , we call J a *fitting cover* of T'_v if the links in J cover all of the tree-edges of T'_v but do not cover any other tree-edge; thus, we have $\cup_{u,w \in J} P'_{u,w} = T'_v$ where $P'_{u,w}$ is the path of tree-edges between nodes u, w in the current tree T' .

7.4.2 Maximum matching

Our algorithm and analysis are based on a *maximum* matching of the leaf-to-leaf links that are neither twin links nor buddy links. Let M denote one such matching; thus, M is a maximum matching of the subgraph $(L, E^{reg}(L))$. By an M -link we mean a link that is in M . We denote the set of M -exposed leaf nodes by U . We will often refer to M and U in the rest of this chapter.

For the rest of this chapter, unless mentioned otherwise, a semiclosed tree means a subtree that is semiclosed w.r.t. the matching M .

7.4.3 Bad 2-stem trees

Let T_v be a semiclosed tree rooted at v (w.r.t. M) that has exactly 4 leaves and two stems s_1, s_2 , where we denote the leaves of the tree T_{s_i} by u_i, w_i for $i = 1, 2$.

By a *leafy 3-cover* of T_v we mean a set of three links J such that J covers T_v , one of the links in J has one end in T_v and one end in $L - L(T_v)$, and the other two links in J have both ends in T_v .

We call T_v a *bad 2-stem tree* if (i) one of the links in $E(\{u_1, w_1\}, \{u_2, w_2\})$ is in M , (ii) two of the leaves are M -exposed, (iii) one of the leaves is incident to all the links in $E(\{u_1, w_1\}, \{u_2, w_2\})$ (thus $E(\{u_1, w_1\}, \{u_2, w_2\})$ has one or two links), (iv) there exists a cover of T_v of size three, and (v) there exists no leafy 3-cover of T_v . Let us fix the notation such that $w_1 w_2$ is the unique M -link in $E(L(T_v))$; thus, u_1 and u_2 are M -exposed (see Figure 7.4).

By a *maximal bad 2-stem tree* T_v we mean a bad 2-stem tree that is not a proper subtree of another bad 2-stem tree. (Thus, any tree T_q rooted at a proper ancestor q of v (if q exists) must violate one of the conditions for being a bad 2-stem tree.)

Let $\mathcal{F}^{prep} = \{T_{v_1}, \dots, T_{v_k}\}$ denote the set of maximal bad 2-stem trees of T ; clearly, any two trees in \mathcal{F}^{prep} are disjoint, by Property 6.2.1. We use $V(\mathcal{F}^{prep})$ to denote $\bigcup_{T_v \in \mathcal{F}^{prep}} V(T_v)$.

We define $\Lambda = \bigcup_{T_v \in \mathcal{F}^{prep}} L(T_v)$, that is, Λ consists of all the leaves of all the trees in \mathcal{F}^{prep} .

Since each bad 2-stem tree has a cover of size 3, the shadow-closed property implies that each bad 2-stem tree has a fitting cover of size ≤ 3 . Our algorithm applies a preprocessing step that contracts each tree $T_v \in \mathcal{F}^{prep}$ by a fitting cover of size ≤ 3 .

Preprocessing step 1 (Λ -contraction): For every tree $T_v \in \mathcal{F}^{prep}$, add a fitting cover of T_v of size ≤ 3 to F and contract T_v to a compound node.

The “cost” incurred for this step is at most $4|\mathcal{F}^{prep}|$, since each tree in \mathcal{F}^{prep} incurs a cost of ≤ 3 for its fitting cover and a cost of 1 for the resulting compound node.

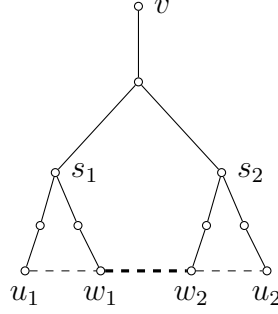


Figure 7.4: Illustration of bad 2-stem tree. The dashed lines indicate links and the thick dashed line indicates an M -link.

This cost is charged to one part of our potential function, namely, it is charged to

$$lbd_y(\Lambda) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl} \cap V(\mathcal{F}^{prep})} y(\delta_E(v)).$$

Lemma 7.4.5 below shows that this quantity is $\geq 4|\mathcal{F}^{prep}|$.

Remark 7.4.2 *The results in this section show that Λ -contraction is valid, in the sense that the algorithm has sufficient credits to pay for the cost of this preprocessing step. Moreover, Λ -contraction is essential for the correctness of the overall algorithm (i.e., the algorithm cannot skip Λ -contraction), because the proof of Theorem 7.6.6 (Subcase 2.2) relies on Λ -contraction.*

Lemma 7.4.3 *Let $\mathcal{F}^{prep} = \{T_{v_1}, \dots, T_{v_k}\}$ denote the set of maximal bad 2-stem trees of T . Then, we have*

$$\begin{aligned} & lbd_y(\Lambda) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl} \cap V(\mathcal{F}^{prep})} y(\delta_E(v)) \\ & \geq \sum_{T_v \in \mathcal{F}^{prep}} \left(\frac{3}{2} y(E(L(T_v))) + \frac{1}{2} y(E(L(T_v), L-L(T_v))) + y(E(L(T_v), V-L)) + \frac{1}{2} \sum_{u \in V(T_v) - L} y(\delta_E(u)) \right) \end{aligned}$$

Proof. Recall that $lbd_y(\Lambda)$ denotes

$$\frac{3}{2} y(E(\Lambda)) + \frac{1}{2} y(E(\Lambda, L-L)) + y(E(\Lambda, V-L)) + \frac{1}{2} \sum_{s \in \mathcal{S}_\Lambda} y(\delta_E(s)).$$

Observe that a bad 2-stem tree has no buds and two stems, so any node of such a tree is either a leaf, or a stem, or a node of \mathcal{R}^{nonspl} . Hence, $(\mathcal{R}^{nonspl} \cap V(\mathcal{F}^{prep})) \cup \mathcal{S}_\Lambda = \bigcup_{T_v \in \mathcal{F}^{prep}} (V(T_v) - L)$, and so we have

$$\frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl} \cap V(\mathcal{F}^{prep})} y(\delta_E(v)) + \frac{1}{2} \sum_{s \in \mathcal{S}_\Lambda} y(\delta_E(s)) = \sum_{T_v \in \mathcal{F}^{prep}} \frac{1}{2} \sum_{u \in V(T_v) - L} y(\delta_E(u)).$$

Now, the lemma follows from the following claim.

Claim 7.4.4 *Let $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ be a partition of Λ . Then, we have*

$$\begin{aligned} & \frac{3}{2}y(E(\Lambda)) + \frac{1}{2}y(E(\Lambda, L-\Lambda)) + y(E(\Lambda, V-L)) \\ & \geq \frac{3}{2} \sum_{i=1}^k y(E(\Lambda_i)) + \frac{1}{2} \sum_{i=1}^k y(E(\Lambda_i, L-\Lambda_i)) + \sum_{i=1}^k y(E(\Lambda_i, V-L)). \end{aligned}$$

To prove the claim, observe that

$$\begin{aligned} & \frac{3}{2}y(E(\Lambda)) + \frac{1}{2}y(E(\Lambda, L-\Lambda)) + y(E(\Lambda, V-L)) \\ & \geq \left(\frac{3}{2} \sum_{i=1}^k y(E(\Lambda_i, \Lambda_i)) + \frac{1}{2} \sum_{i=1}^k \sum_{j \neq i} y(E(\Lambda_i, \Lambda_j)) \right) + \frac{1}{2} \sum_{i=1}^k y(E(\Lambda_i, L-\Lambda)) + \sum_{i=1}^k y(E(\Lambda_i, V-L)) \\ & = \frac{3}{2} \sum_{i=1}^k y(E(\Lambda_i)) + \frac{1}{2} \sum_{i=1}^k y(E(\Lambda_i, L-\Lambda_i)) + \sum_{i=1}^k y(E(\Lambda_i, V-L)). \end{aligned}$$

We partition Λ into the sets $\Lambda_i = L(T_{v_i})$, where T_{v_1}, \dots, T_{v_k} are the maximal bad 2-stem trees in \mathcal{F}^{prep} . Then, the lemma follows. \square

Lemma 7.4.5 *Let T_v be a bad 2-stem tree. Then we have*

$$\frac{3}{2}y(E(L(T_v))) + \frac{1}{2}y(E(L(T_v), L-L(T_v))) + y(E(L(T_v), V-L)) + \frac{1}{2} \sum_{u \in V(T_v) - L} y(\delta_E(u)) \geq 4.$$

Hence, we have

$$lbd_y(\Lambda) + \frac{1}{2} \sum_{v \in \mathcal{R}^{nonspl} \cap V(\mathcal{F}^{prep})} y(\delta_E(v)) \geq 4|\mathcal{F}^{prep}|.$$

Proof. The second statement follows immediately from the first statement and Lemma 7.4.3. We focus on the first statement.

Let s_1, s_2 denote the two stems in T_v , and let u_1, w_1 (u_2, w_2 , respectively) denote the two leaves in T_{s_1} (T_{s_2} , respectively). Thus, $L(T_v) = \{u_1, w_1, u_2, w_2\}$. W.l.o.g. let $w_1 w_2$ denote the unique M -link in $E(L(T_v))$; the two twin links $u_1 w_1, u_2 w_2$ are also in $E(L(T_v))$, and also $E(L(T_v))$ may contain one other link incident to w_1 or w_2 ; there is no link between u_1 and u_2 (see Figure 7.4). Note that every link incident to u_1 or u_2 (the M -exposed leaves of T_v) must have both ends in T_v , since T_v is semiclosed w.r.t. M .

Let J denote the set of links incident to the leaves of T_v , thus, $J = \bigcup_{i=1,2} (\delta_E(u_i) \cup \delta_E(w_i))$. For any feasible solution x of LP7, Lemmas 6.4.3 implies that $|\text{ones}(x) \cap J| \leq 4$. Hence, by Theorem 6.4.1, y can be written as a convex combination $\sum_{i \in Z} \lambda_i x^i$ such that $x^i \in \text{Las}_{proj}^3(\text{LP7})$ and $x^i|_J$ is integral, $\forall i \in Z$. Note that we can apply Lemma 7.2.2 to x^i since $x^i \in \text{Las}_{proj}^3(\text{LP7})$.

Thus, it suffices to show that for any $i \in Z$, we have

$$\frac{3}{2}x^i(E(L(T_v))) + \frac{1}{2}x^i(E(L(T_v), L-L(T_v))) + x^i(E(L(T_v), V-L)) + \frac{1}{2} \sum_{u \in V(T_v)-L} x^i(\delta_E(u)) \geq 4;$$

let α denote the left-hand side of the above inequality.

Observe that every link in J with positive x^i -value must have x^i -value one. Suppose that one of the links $\ell \in E(L(T_v))$ has positive x^i -value, thus $x^i(\ell) = 1$; then we have $\frac{3}{2}x^i(E(L(T_v))) \geq \frac{3}{2}$, thus ℓ contributes value $\frac{3}{2}$ to α . Also, if one of the links ℓ with one end in $L(T_v)$ and the other end at a non-leaf node of T_v has positive x^i -value, then we have $x^i(E(L(T_v), V-L)) + \frac{1}{2} \sum_{u \in V(T_v)-L} x^i(\delta_E(u)) \geq 1 + \frac{1}{2} = \frac{3}{2}$, thus, ℓ contributes value $\frac{3}{2}$ to α .

We complete the proof by case analysis, by considering the number of links with positive x^i -value such that one end is in $L(T_v)$ and another end is not in T_v .

Case 1. Suppose that there are no links with positive x^i -value such that one end is in $L(T_v)$ and another end is not in T_v . Then, we focus on the number of links in $E(L(T_v))$ that have positive x^i -value; this number is zero, one, or two, because every link incident to a leaf has x^i -value zero or one, and moreover, $x^i(\delta_E(u)) = 1$ for each leaf u of T_v , by LP7 and Lemma 6.4.3. Thus, we have three subcases.

Subcase 1.1. $x^i(E(L(T_v))) = 0$. Every link of x^i -value one incident to a leaf of T_v has its other end at a non-leaf node of T_v , hence, each such link contributes $\frac{3}{2}$ to α ; we have four such links, so $\alpha \geq 6$.

Subcase 1.2. $x^i(E(L(T_v))) = 1$. The link ℓ in $E(L(T_v))$ with x^i -value one contributes $\frac{3}{2}$ to α . The two leaves in $L(T_v)$ that are not incident to ℓ each contribute $\frac{3}{2}$ to α , because each is incident to a link with x^i -value one that has its other end at a non-leaf node of T_v , hence, $\alpha \geq \frac{9}{2}$.

Subcase 1.3. $x^i(E(L(T_v))) = 2$. Since $x^i(\delta_E(u)) = 1$ for each leaf u of T_v , the links in $E(L(T_v))$ with x^i -value one share no ends. By the definition of bad 2-stem trees, all the links in $E(\{u_1, w_1\}, \{u_2, w_2\})$ are incident to one of the four leaves. It follows that the twin links u_1w_1 and u_2w_2 both have x^i -value one, and each contributes $\frac{3}{2}$ to α . Next, focus on the stems in T_v ; by Lemma 7.2.2, $x^i(\delta_E^{out}(s_j)) \geq x^i(u_jw_j) = 1$ for $j = 1, 2$. Thus, each stem contributes at least $\frac{1}{2}$ to α via the term $\frac{1}{2} \sum_{u \in V(T_v) - L} x^i(\delta_E(u))$. Hence, $\alpha \geq 2(\frac{3}{2}) + 2(\frac{1}{2}) = 4$.

Case 2. Suppose that there is at least one link with positive x^i -value such that one end is in $L(T_v)$ and another end is not in T_v . Again, we focus on the number of links in $E(L(T_v))$ that have positive x^i -value; it can be seen that this number is zero or one. (Note that every link incident to a leaf has x^i -value zero or one, and moreover, for every leaf u of T_v , we have $x^i(\delta_E(u)) = 1$.) Thus, we have two subcases.

Subcase 2.1. $x^i(E(L(T_v))) = 0$. Every link of x^i -value one incident to u_1 or u_2 (the M -exposed leaves of T_v) has its other end at a non-leaf node of T_v , hence, each such link contributes $\frac{3}{2}$ to α ; we have two such links. Every link of x^i -value one incident to w_1 or w_2 (the M -covered leaves of T_v) has its other end either in $V-L$, and so contributes 1 to $x^i(E(L(T_v), V-L))$, or else has its other end in $L-L(T_v)$, and so contributes $\frac{1}{2}$ to $\frac{1}{2}x^i(E(L(T_v), L-L(T_v)))$; again, we have two such links. Hence, $\alpha \geq 2(\frac{3}{2}) + 2(\frac{1}{2}) = 4$.

Subcase 2.2. $x^i(E(L(T_v))) = 1$. Let ℓ_v denote a link of positive x^i -value such that one end is in $L(T_v)$ and another end is not in T_v . Note that $x^i(\ell_v) = 1$. The end of ℓ_v in $L(T_v)$ must be M -covered; w.l.o.g. assume that this node is w_1 ; let q be the end of ℓ_v in $V-V(T_v)$.

Note that $x^i(\delta_E(u)) = 1$ for each leaf u of T_v . By definition of bad 2-stem tree T_v , the links in $E(L(T_v))$ that are not incident to w_1 are the twin link u_2w_2 and possibly the link u_1w_2 (there is no link between u_1, u_2). Thus, either $x^i(u_2w_2) = 1$ or $x^i(u_1w_2) = 1$, and this contributes $\frac{3}{2}$ to α via the term $\frac{3}{2}x^i(E(L(T_v)))$. In either case, one of the M -exposed leaves u_1 or u_2 is incident to a link of x^i -value one that has its other end at a non-leaf node of T_v , and this contributes $\frac{3}{2}$ to α via the term $x^i(E(L(T_v), V-L)) + \frac{1}{2} \sum_{u \in V(T_v) - L} x^i(\delta_E(u))$.

We have two subcases, depending on whether q is a leaf or not.

Subcase 2.2.1. $q \notin L$. Then, the link $\ell_v = w_1q$ contributes 1 to α via the term $x^i(E(L(T_v), V-L))$. Then, we have $\alpha \geq \frac{3}{2} + \frac{3}{2} + 1 = 4$.

Subcase 2.2.2. $q \in L$. Then, the link $\ell_v = w_1q$ contributes $\frac{1}{2}$ to α via the term $\frac{1}{2}x^i(E(L(T_v), L-L(T_v)))$.

Moreover, observe that we cannot have $x^i(u_1w_2) = 1$, otherwise the three links $\ell_v = w_1q, u_1w_2, u_2w_2$ form a leafy 3-cover of T_v (see Figure 7.5(a)).

Thus, u_2w_2 is the unique link in $E(L(T_v))$ with x^i -value one. Now, focus on the stem s_2 ; by Lemma 7.2.2, $x^i(\delta_E^{out}(s_2)) \geq x^i(u_2w_2) = 1$. Thus, $\delta_E^{out}(s_2)$ contributes at least $\frac{1}{2}$ to α via the term $\frac{1}{2} \sum_{u \in V(T_v) - L} x^i(\delta_E(u))$. Finally, note that u_1s_2 is not present; otherwise, we would have a leafy 3-cover consisting of the (hypothetical) link u_1s_2 and the links $\ell_v = w_1q, u_2w_2$ (see Figure 7.5(b)). So, the link of x^i -value one incident with u_1 is not in $\delta_E^{out}(s_2)$. Thus, we have $\alpha \geq \frac{3}{2} + \frac{3}{2} + \frac{1}{2} + \frac{1}{2} = 4$.

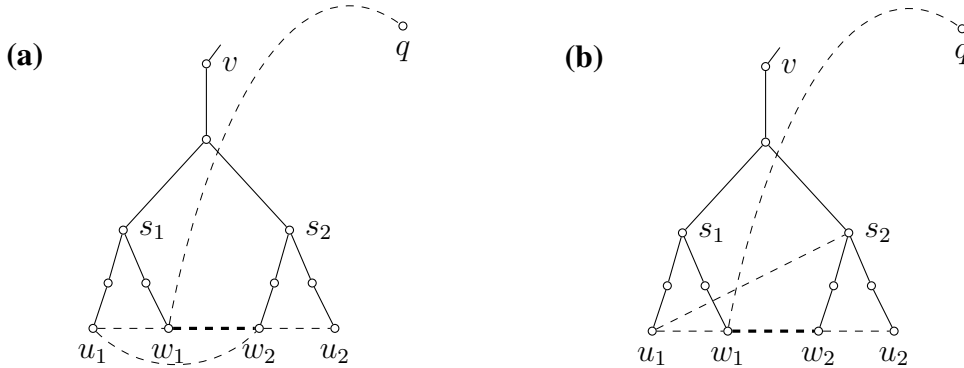


Figure 7.5: Illustration of leafy 3-covers of T_v for Subcase 2.2.2 in the proof of Lemma 7.4.5. The dashed lines indicate links and the thick dashed lines indicate M -links.

This completes the case analysis, and completes the proof. \square

7.4.4 Credit assignment for the algorithm and the preprocessing

Recall that the algorithm starts with a number of credits equal to the potential function, namely the right-hand side of the inequality in Lemma 7.3.3. In order to specify the potential function, we need to specify $\widehat{M}_{(L-\Lambda)}^{reg}, \widehat{U}_{(L-\Lambda)}$. We take $\widehat{M}_{(L-\Lambda)}^{reg}$ to be $M \cap E(L-\Lambda)$, i.e., the restriction

of M to the subgraph $(L-\Lambda, E^{reg}(L-\Lambda))$. It can be seen that $M \cap E(L-\Lambda) = M - E(\Lambda)$ is a maximum matching of the subgraph $(L-\Lambda, E^{reg}(L-\Lambda))$, as required by the definition of $\widehat{M}_{(L-\Lambda)}^{reg}$ in Section 7.3. We take $\widehat{U}_{(L-\Lambda)}$ to be $U \cap (L-\Lambda)$; again, this agrees with the definition of $\widehat{U}_{(L-\Lambda)}$ in Section 7.3.

The “cost” of Λ -contraction (the preprocessing for bad 2-stem trees) is

$$\leq lbd_y(\Lambda) + \frac{1}{2} \sum_{u \in \mathcal{R}^{nonspl} \cap V(\mathcal{F}^{prep})} y(\delta_E(u)),$$

where \mathcal{F}^{prep} is defined above. We subtract this quantity from our potential function and then plus the credits assigned to the compound nodes formed by contracting maximal bad 2-stem trees in \mathcal{F}^{prep} , to get the remaining potential function (total credits available) for the rest of the execution. Clearly, the remaining potential function is

$$\begin{aligned} & \frac{3}{2} |M \cap E(L-\Lambda)| + |U \cap (L-\Lambda)| + |\mathcal{F}^{prep}| + \\ & \frac{1}{2} y(E(V-L)) + \frac{1}{2} \sum_{u \in (\mathcal{R}^{nonspl} - V(\mathcal{F}^{prep}))} y(\delta_E(u)) + \frac{1}{2} \sum_{u \in \mathcal{R}^{special}} y(E(u, V-L_{(L-\Lambda)}^{bud}(u))) + \sum_{s \in \mathcal{S} - \mathcal{S}_\Lambda} slack_y(s) \end{aligned}$$

By the *integral potential function* we mean the sum of the first three terms above (namely, $\frac{3}{2} |M \cap E(L-\Lambda)| + |U \cap (L-\Lambda)| + |\mathcal{F}^{prep}|$), and by the *fractional potential function* we mean the sum of the remaining terms (namely, the sum of the four terms that use y).

The following observation is useful for simplifying our notation.

Fact 7.4.6 *For the current tree T' (after Λ -contraction), $U \cap (L-\Lambda)$ is the same as the set of M -exposed original leaves, and $M \cap E(L-\Lambda) = M(T')$.*

We start with the credit given by the integral potential function, and we maintain the following assignment of credits to the nodes of T' and the links of $M(T') = M \cap E(L-\Lambda)$:

- every M -exposed original leaf has one credit,
- every compound node has one credit,
- every link of $M(T')$ has $\frac{3}{2}$ credit, and
- the root r has one credit.

It can be seen that the integral potential function suffices for assigning credits to the tree T' that results from Λ -contraction. (See Section 6.6, for a discussion on the unit credit for the root r .)

We define the *integral credit* of a set of links J (w.r.t. T') to be the sum of the (integral) credits of the M -links pq such that $V(P'_{p,q}) \subseteq \bigcup_{uw \in J} V(P'_{u,w})$, plus the sum of the (integral) credits of the nodes in $(\bigcup_{uw \in J} V(P'_{u,w})) - (\mathcal{S} \cup \mathcal{R})$, plus one if r occurs as an original node in $\bigcup_{uw \in J} P'_{u,w}$. In other words, the integral credit of J is the sum of $\frac{3}{2}$ times the number of M -links pq such that $V(P'_{p,q}) \subseteq \bigcup_{uw \in J} V(P'_{u,w})$, plus the number of compound nodes in $(\bigcup_{uw \in J} V(P'_{u,w})) - (\mathcal{S} \cup \mathcal{R})$, plus the number of M -exposed original leaves in the same set, plus one if r occurs as an original node in $\bigcup_{uw \in J} P'_{u,w}$. Informally speaking, this is the amount of integral credit “released” when we contract all the links in J (these credits are available at this step, but are not available later on in the execution).

Now, consider the fractional potential function. We use it to maintain an assignment of *fractional credits* to the (rooted) subtrees of T' .

For any subtree T'_v of T' , observe that $V(T'_v) \cap (\mathcal{R}^{\text{nonspl}} - V(\mathcal{F}^{\text{prep}})) = V(T'_v) \cap \mathcal{R}^{\text{nonspl}}$, because none of the original nodes in $V(\mathcal{F}^{\text{prep}})$ is present in T' after the Λ -contraction; similarly, we have $V(T'_v) \cap (\mathcal{S} - \mathcal{S}_\Lambda) = V(T'_v) \cap \mathcal{S}$, because none of the stems in \mathcal{S}_Λ is present in T' after the Λ -contraction.

For any subtree T'_v of T' and for any vector $x \in \mathbb{R}^E$, we use $\Phi(x, T'_v)$ to denote

$$\begin{aligned} & \frac{1}{2} x(E((V-L) \cap V(T'_v), (V-L) \cap (V(T') - V(T'_v)))) + \\ & \frac{1}{2} \sum_{u \in V(T'_v) \cap \mathcal{R}^{\text{nonspl}}} x(\delta_E(u)) + \frac{1}{2} \sum_{u \in V(T'_v) \cap \mathcal{R}^{\text{special}}} x(E(u, V - L_{(L-\Lambda)}^{\text{bud}}(u))) + \sum_{s \in V(T'_v) \cap \mathcal{S}} \text{slack}_x(s). \end{aligned}$$

Informally speaking, $\Phi(y, T'_v)$ denotes the fractional credit of T'_v , that is, the part of the fractional potential function that is “owned” by T'_v . This fractional credit of T'_v will be used together with its integral credit for contracting T'_v . The first term is defined on the set of links with ends that are original in T' such that one is in T'_v and the other is not in T'_v . After contracting the subtree T'_v , one end of each link in this set would become a compound node in the current tree. Thus, the credits of the first term are used only once. Similarly, the last three terms are credits that assigned to original nodes in T'_v . These credits are also used only once.

7.4.5 Second preprocessing step

We apply a second preprocessing step after the Λ -contraction and before the main loop of the algorithm.

Let b_0 be a bud in T_s rooted at a stem s , and let b_1b_2 be the buddy link of b_0 , where the leaves of T_s are b_0, b_1 . Moreover, if b_1 is also a bud, then we always assume w.l.o.g. that $up(b_0)$ is an ancestor of $up(b_1)$.

Preprocessing step 2: If all three of b_0, b_1, b_2 are M -exposed, then we contract the two links $up(b_0)b_0$ and b_1b_2 . Repeat this procedure until no such M -exposed b_0, b_1, b_2 are applicable.

Fact 7.4.7 *Preprocessing step 2 has sufficient credits.*

Proof. Note that each of the M -exposed leaves b_0, b_1, b_2 has 1 unit of credit. Thus, we have sufficient credit for contracting two links and forming one new compound node. \square

Remark 7.4.8 *The results in this section show that this step is valid, in the sense that the algorithm has sufficient credits to pay for the cost of this step. Additionally, Preprocessing step 2 contracts the node set that is disjoint with that contracted by Preprocessing step 1. Moreover, Preprocessing step 2 is also essential for the correctness of the overall algorithm (i.e., the algorithm cannot skip this step), because the proof of Lemma 7.6.2 relies on this step.*

7.5 Algorithm and credits II: Overall algorithm

We first discuss two key concepts for the algorithm, and then we present pseudo-code for the overall algorithm. Also, we state and prove several assertions, i.e., we prove some basic properties maintained by the algorithm. These assertions are critical for the analysis in the next section.

7.5.1 (Up-to-5) greedy contractions and assertions on M

Recall that the integral credit of a set of links J is the sum of the (integral) credits of the M -links pq such that $V(P'_{p,q}) \subseteq \bigcup_{uw \in J} V(P'_{u,w})$, plus the sum of the (integral) credits of the nodes in $(\bigcup_{uw \in J} V(P'_{u,w})) - (\mathcal{S} \cup \mathcal{R})$, plus one if r occurs as an original node in $\bigcup_{uw \in J} P'_{u,w}$.

We define an *(up-to-5) greedy contraction* to be a contraction of a set of links J such that

- (i) $|J| \leq 5$;
- (ii) contraction of J results in a single compound node, i.e., $\bigcup_{uw \in J} P'_{u,w}$ forms a connected graph;
- (iii) the integral credit of J is $\geq |J| + 1$.

Note that an (up-to-5) greedy contraction has sufficient credits by definition. The following assertion is similar to Lemma 6.6.3.

Lemma 7.5.1 (Assertions on M) *Suppose that no (up-to-5) greedy contractions are applicable. Then*

- (1) *For every M -link uw , every node in the path between u and w in T' is an original node. In particular, in T' , both ends of each M -link are original leaf nodes.*
- (2) *There exist no links between M -exposed leaves.*

7.5.2 Good semiclosed trees

Recall that a semiclosed tree means a tree that is semiclosed w.r.t. the matching M , unless mentioned otherwise.

After the preprocessing steps, whenever we mention semiclosed trees, we assume that no (up-to-5) greedy contractions are applicable in the current tree T' . Then, Lemma 7.5.1(1) implies that M is a set of leaf-to-leaf links w.r.t. the current tree T' . Hence, semiclosed trees (w.r.t. M) are well defined.

We use $U(T'_v)$ to denote the set of M -exposed leaves of T'_v (including both original leaf nodes and compound leaf nodes). Let $C(T'_v)$ denote the set of compound non-leaf nodes of T'_v . Also, we use $L^{\text{matched}}(T'_v)$ to denote the set of M -covered leaves of T'_v .

Note that every node in $V(T'_v) \cap \mathcal{R}$, $V(T'_v) \cap \mathcal{S}$ is an original node. Recall that $M(T'_v)$ denotes the set of M -links (w.r.t. T') that have both ends in T'_v .

We define the credit of a (rooted) subtree T'_v of the current tree T' to be the sum of the fractional credit of T'_v , namely, $\Phi(y, T'_v)$, and the integral credit of T'_v . The latter is given by the sum of the following terms: $\frac{3}{2}|M(T'_v)|$, the number of compound nodes in T'_v , the number of M -exposed original leaves in T'_v , an additional one if the root r is in $V(T'_v) \cap \mathcal{R}$. We call a semiclosed tree T'_v *good* if its credit is $\geq |M(T'_v)| + 1$. The potential and credit defined in this chapter are different from that in Chapter 6. However, the next lemma still holds. The proof is similar to the proof of Lemma 6.6.4.

Lemma 7.5.2 *Let T'_v be a semiclosed tree. If at least one of the following conditions is satisfied, then T'_v is good.*

- $T'_v = T'$
- $C(T'_v) \neq \emptyset$
- $|M(T'_v)| \geq 2$
- $\Phi(y, T'_v) \geq 1$
- $|M(T'_v)| = 1$ and $\Phi(y, T'_v) \geq \frac{1}{2}$.

7.5.3 Summary of the algorithm

We start with $F := \emptyset$ (F is the set of links picked by the algorithm) and $T' := T$ (T' is the current tree T/F).

Algorithm 7.1: Find an approximately optimal solution for TAP.

```

1  apply Preprocessing step 1 ( $\Lambda$ -contraction);
2  apply Preprocessing step 2;
3  while  $T'$  is not a single node do
4      repeatedly apply (up-to-5) greedy contractions until no such contractions are
        applicable;
5      find a good semiclosed tree  $T'_v$  with a fitting cover  $J$  of size  $|\Gamma(M, T'_v)|$  (Algorithm 7.2
        in Section 7.6 presents the details for finding such a semiclosed tree);
6      add  $J$  to  $F$ , contract  $T'_v$  to a new compound node, update  $T'$ ;
7  end
```

7.5.4 Stem assertion of the algorithm

This section presents another assertion, called *stem assertion*, that is important for the analysis of the algorithm.

Recall that the algorithm iteratively contracts a set of links such that the tree-paths associated with these links form a connected graph; the set of chosen links and their associated tree-paths are contracted into a new compound node. When we say that a contraction *hits* a node v , we mean

a contraction during the execution contracts a set of links J such that $v \in \bigcup_{uv \in J} V(P'_{u,w})$; thus, at least one of tree-edge incident with v is covered by one of these links and v gets contracted into the compound node formed by that contraction.

Stem assertion: *Let s be a stem. The first contraction that hits a node of T_s (during the execution of the algorithm) must hit s .*

Property 7.5.3 *The algorithm maintains the stem assertion.*

Proof. Let s be a stem. First, consider the two preprocessing steps. Either all of the tree T_s is contracted or none of the tree-edges of T_s is contracted by Preprocessing step 1 (Λ -contraction). The same statement holds for Preprocessing step 2. Hence, the stem assertion is maintained by the two preprocessing steps.

Next, suppose that the first contraction (in the execution) that hits a node in T_s is an (up-to-5) greedy contraction that contracts a set of links J . If one of the links $uw \in J$ has one end in T_s and the other end not in T_s , then the tree path $P'_{u,w}$ (in the current tree considered at the moment) must contain s , hence, the stem assertion is maintained. The remaining possibility is that all links of J have both ends in T_s . Then observe that the number of integral credits available in T_s is either zero, one, or two, and, in the last case, both leaves of T_s are M -exposed. (Note that the root r is a proper ancestor of s in T , by definition of stem.) The greedy contraction of J requires $|J| + 1$ integral credits. Thus, $|J| = 1$ and J contains the twin link of s . Contracting the twin link clearly maintains the stem assertion.

Finally, suppose that the first contraction (in the execution) that hits a node in T_s is the contraction of a good semiclosed tree T'_v . By Property 6.2.1, one of $T_s = T'_s$, T'_v is contained in the other. It is easily seen that no proper (rooted) subtree of T_s is a semiclosed tree. (Note that there is a shadow link between each leaf and s via the twin link of s , hence, any semiclosed tree containing a leaf of T_s that is M -exposed will contain s too. Also, any semiclosed tree containing a leaf of T_s that is M -covered will contain s too.) The only remaining possibility is that T'_v contains T_s ; then, the contraction of T'_v maintains the stem assertion. \square

7.6 Analysis of the algorithm, and deficient trees

This section has our main results. Informally speaking, the key result (Theorem 7.6.6) asserts the following: if a semiclosed tree T'_v is not good, then either T'_v is a deficient tree (defined below) or T'_v is a particular type of tree that is easily bypassed by our analysis.

The analysis consists of two parts. In Section 7.6.2, using local integrality of feasible solutions to the Lasserre system, we show that all semiclosed trees are good, except for a few cases. The nontrivial cases give deficient trees. Section 7.6.3 shows how to handle deficient trees. This leads to an efficient algorithm for finding a good semiclosed tree together with a fitting cover of appropriate size.

Recall the definition of deficient 3-leaf tree from Section 6.7.

Deficient 3-leaf tree: Suppose that T'_v is a semiclosed tree with exactly three leaves a, b_1, b_2 . Clearly, among the nodes w of T'_v either there is exactly one node with $\deg_{T'}(w) = 4$ or there are two nodes with degree 3 in T' . In the latter case, we denote these two nodes by u and q ; moreover, we fix the notation such that u is an ancestor of q , and the leaf b_1 is not a descendant of q ; thus, a, b_2 (but not b_1) are descendants of q . In the former case, we denote by u the unique node that is incident to four tree-edges. We call T'_v a *deficient 3-leaf tree* (see Figure 7.6) if (i) the link b_1b_2 is present and it is in M , (ii) the link ab_1 is present, and (iii) there exists a link b_2w such that $w \in V(T') - V(T'_v)$.

Moreover, in the first case (with a unique node u in T'_v with $\deg_{T'}(u) = 4$), if conditions (i)–(iii) hold with both labelings (b_1, b_2) and (b_2, b_1) of the M -link, then we fix the notation such that $up(b_2)$ is an ancestor of $up(b_1)$. For both cases, we call b_2 the *ceiling leaf* of T'_v .

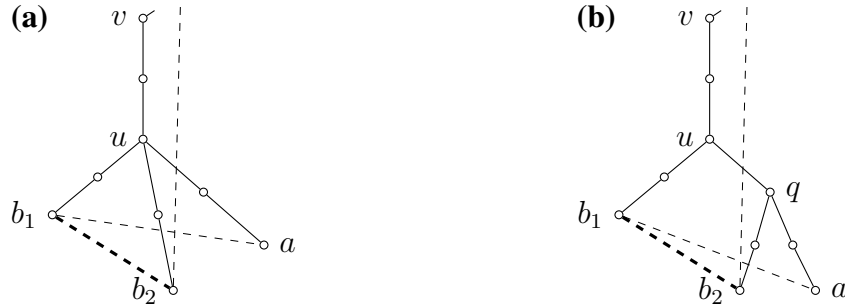


Figure 7.6: Illustration of deficient 3-leaf tree. The dashed lines indicate links and the thick dashed lines indicate M -links.

Deficient 4-leaf tree: Let T'_v be a semi-closed tree with 4 leaves a, b_1, b_2, c , and exactly one stem node s , and exactly one M -link such that all nodes in T'_s are original, the leaves of T'_s are a, b_1 , and the M -link is b_1b_2 . Let p be the least common ancestor of s and c . Then, T'_v is called a *deficient 4-leaf tree* (see Figure 7.7) if (i) T'_p contains all leaves of T'_v , (ii) the link cs is present, and (iii) there exists a link b_2w such that $w \in V(T') - V(T'_v)$. We call the link cs the *latch* of T'_v .

The contraction of the latch cs in a deficient 4-leaf tree results in a deficient 3-leaf tree due to

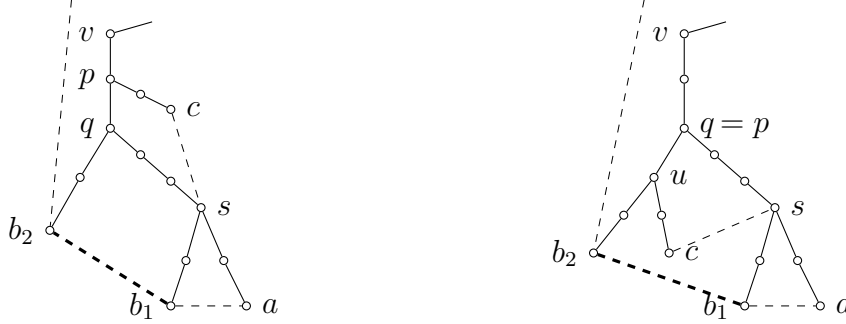


Figure 7.7: Illustration of deficient 4-leaf trees. The dashed lines indicate links and the thick dashed lines indicate M -links.

the presence of the links $ab_1, up(b_2)b_2$ (see Figure 7.6(a)). Let b be the ceiling leaf of the resulting tree. Clearly, $up(b)$ is an ancestor of $up(b_2)$.

7.6.1 Properties from assertions

The next lemma states some properties pertaining to stem nodes and semiclosed trees; these properties are often used in the analysis of the algorithm. The proof of the lemma is based on the stem assertion.

Lemma 7.6.1 *Suppose that no (up-to-5) greedy contractions are applicable. Let T'_v be a semiclosed tree with $C(T'_v) = \emptyset$.*

1. *If T'_v contains a stem node s , i.e., $s \in \mathcal{S} \cap V(T'_v)$, then every node in T'_s is original.*
2. *In the original tree T , suppose that s is a stem node, and w is a leaf of the subtree T_s . If w is contained in a compound node $\langle c \rangle$ that is an M -exposed leaf of T'_v , then all nodes of T_s are contained in $\langle c \rangle$.*
3. *In the original tree T , suppose that s is a stem node, and w is a leaf of the subtree T_s . If w is an original node that is an M -exposed leaf of T'_v , then s is an original node of T'_v .*

Proof. The first part follows from the stem assertion.

Consider the second part. By the stem assertion, and the fact that w is contained in $\langle c \rangle$, it can be seen that s is contained in some compound node. If s is contained in $\langle c \rangle$, then the proof

is done (if a compound leaf node contains a node $u \in V(T)$, then that compound node contains T_u). Now, suppose that s is contained in a different compound node $\langle a \rangle$. Then, there exists a link between $\langle c \rangle$ and $\langle a \rangle$ (in the current tree), because the link ws is present in E (the input) by noting that ws is a shadow of the twin link of T_s . It can be seen that T'_v contains $\langle a \rangle$ as a leaf, because T'_v is semiclosed, $C(T'_v) = \emptyset$, $\langle c \rangle$ is an M -exposed leaf of T'_v , and the link between $\langle c \rangle$ and $\langle a \rangle$ is present. This gives a contradiction because the compound leaf node $\langle a \rangle$ contains s so it contains T_s , hence, $\langle c \rangle$ cannot contain w .

The third part follows from arguments similar to that used for the second part; we give a sketch. Suppose that s is contained in a compound node $\langle a \rangle$. If $\langle a \rangle$ is a leaf of T' , then $\langle a \rangle$ would contain the subtree T_s , and this would contradict the fact that the leaf w is an original node. Since $C(T'_v) = \emptyset$, $\langle a \rangle$ cannot be a non-leaf node of T'_v . Thus, T'_v contains w , but it does not contain $\langle a \rangle$. This contradicts the fact that T'_v is semiclosed, because there is a link between the M -exposed leaf w of T'_v and $\langle a \rangle$ (due to the shadow ws of the twin link incident to w). \square

Lemma 7.6.2 *Suppose that no (up-to-5) greedy contractions are applicable. Let T'_v be a semiclosed tree with $C(T'_v) = \emptyset$. Suppose that one of the M -exposed leaves of T'_v is a compound node $\langle c \rangle$ that contains a bud b_0 , and moreover, there exists an original link $\ell = b_0w$ such that w is not contained in $\langle c \rangle$. Let b_1, b_2 be two nodes in T such that b_1b_2 is the buddy link $\text{buddylk}(b_0)$ and b_0b_1 is a twin link. Then, $\langle c \rangle$ contains b_0, b_1 , it contains no other original leaf nodes, and moreover, $M(T'_v)$ contains a link incident to b_2 .*

Proof. Let s be the stem node associated with the twin link b_0b_1 ; note that the leaves of T_s are b_0, b_1 . By Lemma 7.6.1, T_s is completely contained in $\langle c \rangle$. Since b_0 is contained in an M -exposed leaf $\langle c \rangle$ in T'_v and $C(T'_v) = \emptyset$, we have $\text{up}(b_0)$ is either contained in $\langle c \rangle$ or an original non-leaf node in T'_v . Due to the existence of b_0w , we know that $\text{up}(b_0)$ must be an original non-leaf node in T'_v .

Consider the node b_2 . If b_2 is an original leaf node in T' , since T'_v is a semiclosed tree and there is no link between two M -exposed leaves by Lemma 7.5.1, then the existence of b_1b_2 implies that b_2 is M -covered in T'_v and the other end of the corresponding M -link is also in T'_v . We are done. Otherwise, b_2 is contained in a compound node $\langle a \rangle$. Similarly, we know that $\langle a \rangle$ is a leaf in T'_v . By Lemma 7.5.1, $\langle a \rangle$ cannot be M -covered in T' . Thus, $\langle a \rangle$ must be $\langle c \rangle$ (otherwise, $b_1b_2 \in E$ implies a link between two M -exposed leaves $\langle a \rangle$ and $\langle c \rangle$ in T'). Hence, we can assume that b_0, b_1, b_2 are all contained in $\langle c \rangle$. In what follows, we prove that this contradicts the fact that $\text{up}(b_0)$ is an original non-leaf node in T'_v .

By Preprocessing step 2, one of b_0, b_1, b_2 is M -covered in T (otherwise, the preprocessing step contracts two links to form a compound node that contains $\text{up}(b_0)$).

We have two cases depending on the M -links incident to b_0, b_1, b_2 in T .

- Case 1.** There exists an M -link w.r.t. T incident to one of b_0, b_1, b_2 that has its other end in $L - \{b_0, b_1, b_2\}$; let bu denote such an M -link, where $b \in \{b_0, b_1, b_2\}$, and $u \notin \{b_0, b_1, b_2\}$. Since $\langle c \rangle$ contains all nodes in $\{b_0, b_1, b_2\}$ and bu is an M -link, by Lemma 7.5.1, $V(P_{b,u})$ are contained in $\langle c \rangle$. Note that $V(P_{b,u})$ has an ancestor of $up(b_0)$ in T by the definition of bud. Thus, $up(b_0)$ is contained in $\langle c \rangle$. This is a contradiction.
- Case 2.** Each M -link in T incident to one of b_0, b_1, b_2 has both ends in $\{b_0, b_1, b_2\}$. We can have only one M -link with both ends in $\{b_0, b_1, b_2\}$ in T . Since M has no twin links and no buddy links, b_0b_2 is an M -link, and so b_0 is an M -covered bud in T (see Figure 7.8).

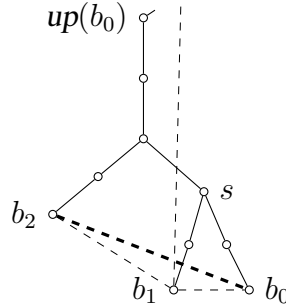


Figure 7.8: Illustration of an M -covered bud in T .

We reach a contradiction by proving the following claim. It completes the proof of this lemma.

Claim. Suppose b_0b_2 is an M -link and b_1 is M -exposed. Then, the first contraction that hits a node of $T_{up(b_0)}$ (during the execution of the algorithm) must hit $up(b_0)$.

If $up(b_1)$ is a descendent of $up(b_0)$, then b_1 is a bud. In this case, b_0b_2 is the buddy link $buddy lk(b_1)$. But b_0b_2 is an M -link. This is a contradiction. Thus, $up(b_1)$ is an ancestor of $up(b_0)$ but not $up(b_0)$. Thus, $up(b_0)$ cannot be r , and any subtree of $T_{up(b_0)}$ is not semiclosed since b_1 is M -exposed.

First, consider the two preprocessing steps. Observe that there are no nodes in $T_{up(b_0)}$ involved in these two steps.

Next, suppose that the first contraction (in the execution) that hits a node in $T_{up(b_0)}$ is an (up-to-5) greedy contraction that contracts a set of links J . If one of the links $uw \in J$ has one end in $T_{up(b_0)}$ and the other end not in $T_{up(b_0)}$, then the tree path $P'_{u,w}$ (in the current tree

considered at the moment) must hit $up(b_0)$, hence, this proves the claim. The remaining possibility is that all links of J have both ends in $T_{up(b_0)}$. Then observe that the number of integral credits available in $T_{up(b_0)}$ is $\frac{5}{2}$ (one M -link plus an M -exposed node b_1). This means the only possible contraction is a contraction of exactly one link. At this moment, there is no compound node in $T_{up(b_0)}$. Thus, an M -link in $T_{up(b_0)}$ only has credit $\frac{3}{2}$. This implies that contraction of an M -link can not be an (up-to-5) greedy contraction. Hence, the only possibility is that this link has two ends at M -exposed leaves. However, $T_{up(b_0)}$ only has one M -exposed leaf.

Finally, suppose that the first contraction (in the execution) that hits a node in $T_{up(b_0)}$ is the contraction of a good semiclosed tree T'_w . By Property 6.2.1, one of $T_{up(b_0)}$, T'_w is contained in the other. Since any subtree of $T_{up(b_0)}$ is not semiclosed, we have T'_w contains $T_{up(b_0)}$. This completes the proof. \square

7.6.2 Most semiclosed trees are good

Let T'_v be a semiclosed tree. Let $L^{matched}(T'_v)$ denote the set of M -covered leaves of T'_v .

We call a compound node $\langle c \rangle$ of T'_v *open* if it is an M -exposed leaf of T'_v , and moreover, $\langle c \rangle$ contains a bud b_0 such that there exists an original link between b_0 and an original node that is not contained in $\langle c \rangle$. We call such a bud an *open bud* (it must be contained in an open compound node and one of the original links incident to the bud is not contained in the compound node). Let $\mathcal{B}^{comp}(T'_v)$ denote the set of open buds of T'_v . Let $\mathcal{B}^{orig}(T'_v)$ denote the set of original nodes of T'_v that are M -exposed buds. Let $\mathcal{B}(T'_v) = \mathcal{B}^{orig}(T'_v) \cup \mathcal{B}^{comp}(T'_v)$.

Lemma 7.6.3 *Suppose that no (up-to-5) greedy contractions are applicable. Let T'_v be a semiclosed tree with $C(T'_v) = \emptyset$, $|M(T'_v)| \leq 1$ and $|\mathcal{S} \cap V(T'_v)| \leq 2$.*

Then, y can be written as a convex combination $\sum_{i \in Z} \lambda_i x^i$ such that $x^i \in \text{Las}_{proj}^3(\text{LP7})$ and $x^i|_J$ is integral, where $J = \{\ell \in \delta_E(u) : u \in (\mathcal{S} \cap V(T'_v)) \cup L^{matched}(T'_v) \cup \mathcal{B}(T'_v)\}$.

Proof. We claim that $|ones(x) \cap J| \leq 14$ for any feasible solution x of LP7. The decomposition of y follows easily from Theorem 6.4.1 and this claim.

To prove the claim, observe that $|\mathcal{S} \cap V(T'_v)| \leq 2$ and $|L^{matched}(T'_v)| = 2|M(T'_v)| \leq 2$, hence, by Lemmas 6.4.3, 7.2.1, we have

$$|ones(x) \cap \{\ell \in \delta_E(u) : u \in \mathcal{S} \cap V(T'_v)\}| \leq 6, \quad |ones(x) \cap \{\ell \in \delta_E(u) : u \in L^{matched}(T'_v)\}| \leq 2,$$

and for each $u \in \mathcal{B}(T'_v)$, we have $|\text{ones}(x) \cap \delta_E(u)| \leq 1$. Thus, to complete the proof of the claim, we have to show that $|\mathcal{B}(T'_v)| \leq 6$.

First, consider any bud $b_0 \in \mathcal{B}^{\text{orig}}(T'_v)$ and its associated stem s . By Lemma 7.6.1(3), s is an original node of T'_v , and so, $s \in \mathcal{S} \cap V(T'_v)$. By Lemma 7.6.1(1), each leaf of T'_s is an original node in T' . Let b_1 be the leaf of T'_s other than b_0 . By Lemma 7.5.1, b_1 is M -covered (otherwise, the twin link b_0b_1 connects two M -exposed leaves). Hence, $b_1 \notin \mathcal{B}^{\text{orig}}(T'_v)$. It follows that each stem in $\mathcal{S} \cap V(T'_v)$ has at most one leaf in $\mathcal{B}^{\text{orig}}(T'_v)$. Since $|\mathcal{S} \cap V(T'_v)| \leq 2$, we have $|\mathcal{B}^{\text{orig}}(T'_v)| \leq 2$.

Now, consider one of the buds $b_0 \in \mathcal{B}^{\text{comp}}(T'_v)$; b_0 is contained in some open compound node and one of the original links incident to b_0 has its other end “outside” this compound node. Then, by Lemma 7.6.2, the buddy link $\text{buddylk}(b_0)$ shares an end with an M -link in $M(T'_v)$. Since $|M(T'_v)| \leq 1$ and each leaf can be an end of at most 2 buddy links, we have $|\mathcal{B}^{\text{comp}}(T'_v)| \leq 4$. Therefore, $|\mathcal{B}(T'_v)| = |\mathcal{B}^{\text{orig}}(T'_v)| + |\mathcal{B}^{\text{comp}}(T'_v)| \leq 6$. This proves the claim. \square

Lemma 7.6.4 *Suppose that no (up-to-5) greedy contractions are applicable. Let T'_v be a semi-closed tree with $C(T'_v) = \emptyset$. Let J be a set of links that each have at least one end in T'_v and no end in $L^{\text{matched}}(T'_v) \cup (\mathcal{S} \cap V(T'_v))$. Let x be a feasible solution of $\text{Las}^3_{\text{proj}}(\text{LP7})$ such that x is integral on the links $\{\delta_E(u) : u \in \mathcal{B}(T'_v)\}$. Then,*

$$\Phi(x, T'_v) \geq \min\left\{\frac{1}{2}, \frac{1}{2}x(J)\right\}.$$

Proof. Let $g(x, T'_v)$ denote $\frac{1}{2} \sum_{u \in \mathcal{R}^{\text{nonspl}} \cap V(T'_v)} x(\delta_E(u)) + \frac{1}{2} \sum_{u \in \mathcal{R}^{\text{special}} \cap V(T'_v)} x(E(u, V - L^{\text{bud}}(u)))$. By Fact 7.3.2, we have $\Phi(x, T'_v) \geq g(x, T'_v)$. We will show that either each link $\ell \in J$ contributes $\frac{1}{2}x(\ell)$ to $g(x, T'_v)$, thereby ensuring $g(x, T'_v) \geq \frac{1}{2}x(J)$, or there exists a set of links that contribute $\frac{1}{2}$ to $g(x, T'_v)$.

Consider any link $\ell \in J$ with $x(\ell) > 0$. By Lemma 7.5.1, ℓ cannot have both ends at M -exposed leaves. Thus, ℓ has at least one end in $V(T'_v) \cap \mathcal{R}$, since it has no end in $L^{\text{matched}}(T'_v) \cup (\mathcal{S} \cap V(T'_v))$ and $C(T'_v) = \emptyset$. It is easily seen that, except for one case, ℓ contributes $\geq \frac{1}{2}x(\ell)$ to $g(x, T'_v)$.

The exceptional case occurs when ℓ has an (original) end at a bud b_0 such that $b_0 \in \mathcal{B}(T'_v)$ and another (original) end w in $\mathcal{R}^{\text{special}}(b_0) \cap V(T'_v)$ by Fact 7.1.1. Then, we have $x(\ell) = 1$ because x is integral on the links incident with nodes in $\mathcal{B}(T'_v)$. We claim that $g(x, T'_v) \geq \frac{1}{2}$. Let \hat{e} denote the tree-edge between w (the end of ℓ in $V(T'_v) \cap \mathcal{R}$) and its parent. Let $\delta^+(\hat{e})$ denote the set of links with positive x -value that cover \hat{e} . Clearly, $x(\delta^+(\hat{e})) \geq 1$. Consider any link $\ell_q = qu$ that is in $\delta^+(\hat{e})$, where q is a descendent of w (possibly, $q = w$) and u is not in T'_w . If q is a

proper descendant of w , then (by the definition of $\mathcal{R}^{special}(b_0)$) q is a descendant of the unique child of w (see Figure 7.3), hence, ℓ and ℓ_q form an overlapping pair such that $x(\ell) + x(\ell_q) > 1$; this is impossible, by the overlapping pair constraints. Hence, we have $q = w$. Clearly, u (the other end of ℓ_q) cannot belong to $L^{bud}(w)$ since u is not a descendant of q . Hence, we have $g(x, T'_v) \geq \frac{1}{2}x(E(w, V - L^{bud}(w))) \geq \frac{1}{2}x(\delta^+(\hat{e})) \geq \frac{1}{2}$. This completes the proof. \square

Let T'_v be a semiclosed tree. We construct an auxiliary graph in order to analyze the credits available in T'_v . We denote the auxiliary graph by $AG(T'_v)$. This is a bipartite graph, and the two sets in the node bipartition are denoted by $ML(T'_v)$ and $AU(T'_v)$. The first set consists of the M -covered leaves $L^{matched}(T'_v)$ and the stems $\mathcal{S} \cap V(T'_v)$. The second set contains an auxiliary node \bar{v} (informally speaking, \bar{v} represents the node set $V(T') - V(T'_v)$), as well as all the M -exposed leaves of T'_v , thus, $AU(T'_v) = \{\bar{v}\} \cup U(T'_v)$.

We define the edge set of $AG(T'_v)$ as follows: for every link pq w.r.t. T' with $p \in ML(T'_v)$, $q \in U(T'_v)$, the edge pq is in $AG(T'_v)$, and for every link pq w.r.t. T' such that $p \in ML(T'_v)$, $q \in V(T') - V(T'_v)$, the edge $p\bar{v}$ is in $AG(T'_v)$. Thus, $AG(T'_v)$ is a multigraph (multiple copies of an edge may be present), and every edge in $AG(T'_v)$ corresponds to a link w.r.t. T' (see Figure 7.9).

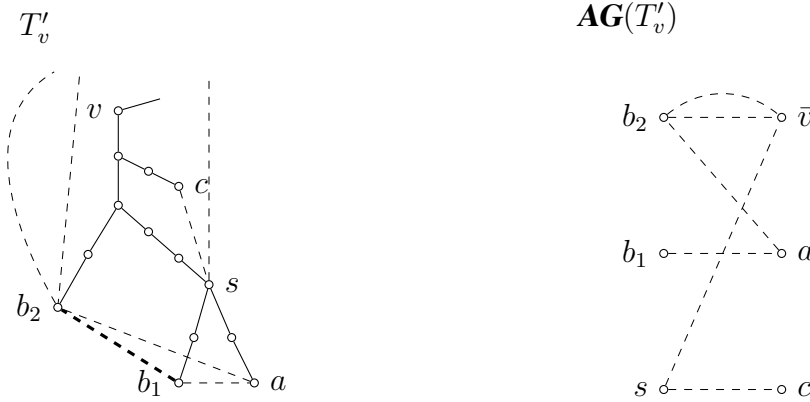


Figure 7.9: Illustration of auxiliary graph. The left figure shows a semiclosed tree T'_v where the dashed lines indicate links and the thick dashed line indicates an M -link. The right figure shows the auxiliary graph $AG(T'_v)$ and its node bipartition $\{s, b_1, b_2\}$, $\{\bar{v}, a, c\}$.

In what follows, we may abuse the notation by not distinguishing between edges (sets of edges) of $AG(T'_v)$ and the corresponding links (sets of links).

Lemma 7.6.5 *Suppose that no (up-to-5) greedy contractions are applicable. Let T'_v be a semiclosed tree such that $T'_v \neq T'$, $C(T'_v) = \emptyset$, $|M(T'_v)| \leq 1$ and $|\mathcal{S} \cap V(T'_v)| \leq 2$.*

1. If $M(T'_v) = \emptyset$, then for any feasible solution $x \in \text{Las}_{proj}^3(\text{LP7})$, we have $\Phi(x, T'_v) \geq 1$. Furthermore, T'_v is good.
2. Suppose that $|M(T'_v)| = 1$, and $|U(T'_v)| \geq |\mathcal{S} \cap V(T'_v)| + 1$. Moreover, suppose that x is a feasible solution in $\text{Las}_{proj}^3(\text{LP7})$ such that $x|_J$ is integral and $\Phi(x, T'_v) < \frac{1}{2}$, where $J = \{\ell \in \delta_E(u) : u \in (\mathcal{S} \cap V(T'_v)) \cup L^{\text{matched}}(T'_v) \cup \mathcal{B}(T'_v)\}$.
Then, $|U(T'_v)| = |\mathcal{S} \cap V(T'_v)| + 1$. Moreover, the auxiliary graph has a perfect matching $\text{AM}(T'_v)$ such that the following conditions hold (for the set of links corresponding to $\text{AM}(T'_v)$):
 - (i) $x(\ell) = 1$ for each link $\ell \in \text{AM}(T'_v)$,
 - (ii) the links of $\text{AM}(T'_v)$ cover T'_v ,
 - (iii) for each stem node $s \in \mathcal{S} \cap V(T'_v)$, $\text{twinkl}(s)$ is in $\text{AM}(T'_v)$,
 - (iv) $\text{AM}(T'_v)$ has no links of the form $\bar{v}s$, where $s \in \mathcal{S} \cap V(T'_v)$.

Proof. Let \hat{e}_v denote the tree-edge between v and its parent; \hat{e}_v is well defined since $T'_v \neq T'$. Let $\bar{J} = \delta_E(\hat{e}_v) \cup (\cup_{u \in U(T'_v)} \delta_E(u))$. Then, $x(\bar{J}) = x(\delta_E(\hat{e}_v)) + \sum_{u \in U(T'_v)} x(\delta_E(u)) \geq 1 + |U(T'_v)|$; the equation holds because (i) T'_v is semiclosed so none of the links in $\delta_E(\hat{e}_v)$ is incident to an M -exposed leaf of T'_v , and (ii) by Lemma 7.5.1(2), no link has both ends at M -exposed leaves; the inequality holds because $x(\delta_E(\hat{e})) \geq 1$ for every tree-edge \hat{e} .

Let $g(x, T'_v)$ denote $\frac{1}{2} \sum_{u \in \mathcal{R}^{\text{nonspec}} \cap V(T'_v)} x(\delta_E(u)) + \frac{1}{2} \sum_{u \in \mathcal{R}^{\text{special}} \cap V(T'_v)} x(E(u, V - L^{\text{bud}}(u)))$.

By Fact 7.3.2, we have $\Phi(x, T'_v) \geq g(x, T'_v)$.

Part (1) $M(T'_v) = \emptyset$. Clearly, $|U(T'_v)| \geq 1$, since T'_v has at least one leaf.

Note that $\frac{1}{2}x(\bar{J}) \geq \frac{1}{2}(1 + |U(T'_v)|) \geq 1$. We will show that every link $\ell \in \bar{J}$ contributes $\geq \frac{1}{2}x(\ell)$ to $g(x, T'_v)$.

First, we show that $\mathcal{S} \cap V(T'_v) = \emptyset$. Otherwise, consider any $s \in \mathcal{S} \cap V(T'_v)$; by Lemma 7.6.1(1), every node in T'_s is original, and so the twin link of s has both ends at M -exposed nodes (since $M(T'_v) = \emptyset$); this is impossible due to Lemma 7.5.1.

Next, we show that $\mathcal{B}(T'_v) = \emptyset$. Since $\mathcal{S} \cap V(T'_v) = \emptyset$, Lemma 7.6.1(3) implies that $\mathcal{B}^{\text{orig}}(T'_v) = \emptyset$ (if a bud $b_0 \in \mathcal{B}^{\text{orig}}(T'_v)$ is present as an original M -exposed leaf of T'_v , then its stem is present as an original node of T'_v). Also, we have $\mathcal{B}^{\text{comp}}(T'_v) = \emptyset$, otherwise, by Lemma 7.6.2, there exists an M -link between two leaves in T'_v (this is impossible since $M(T'_v) = \emptyset$). Thus, $\mathcal{B}(T'_v) = \emptyset$.

Now, observe that every link $\ell \in \bar{J}$ has an end in $V(T'_v) \cap \mathcal{R}$, and hence, it contributes $\frac{1}{2}x(\ell)$ to $g(x, T'_v)$. Therefore, $\Phi(x, T'_v) \geq 1$. Since this inequality holds for every feasible solution x in $\text{Las}_{proj}^3(\text{LP7})$, it also holds for the optimal solution y of $\text{Las}_{proj}^t(\text{LP7})$ for $t \geq 17$; see Theorem 7.4.1. Thus, we have $\Phi(y, T'_v) \geq 1$. Hence, by Lemma 7.5.2, T'_v is good.

Part (2) In this case, $|M(T'_v)| = 1$, $|U(T'_v)| \geq |\mathcal{S} \cap V(T'_v)| + 1$. Moreover, $x|_J$ is integral, where $J = \{\ell \in \delta_E(u) : u \in (\mathcal{S} \cap V(T'_v)) \cup L^{\text{matched}}(T'_v) \cup \mathcal{B}(T'_v)\}$.

First, we show that $x(\delta_E(s)) \leq 1$ for each stem $s \in \mathcal{S} \cap V(T'_v)$. We use a contradiction argument. Suppose that $x(\delta_E(s)) > 1$. Let $\delta_E^+(s)$ denote the set of links of positive x -value incident to s . Note that every link ℓ in $\delta_E^+(s)$ has $x(\ell) = 1$ because $x|_J$ is integral. Since $x(\delta_E(s)) > 1$, we have $|\delta_E^+(s)| \geq 2$. This implies that each of the buddy links associated with the stem s has x -value zero. To see this, note that the set of links in $\delta_E^+(s)$ covering each of the three tree-edges incident to s is an overlapping clique, hence, at most one link of $\delta_E^+(s)$ belongs to one of these overlapping cliques (by the overlapping pair constraints); moreover, each buddy link associated with s is overlapping with the links that belong to two of these overlapping cliques (see Figure 7.10); hence, if a buddy link ℓ associated with s has $x(\ell) > 0$, then we get a violation (of an overlapping pair constraint) for one of the three overlapping cliques. Since the buddy links associated with s (if any) have x -value zero, we have $\Phi(x, T'_v) \geq \text{slack}_x(s) \geq \frac{1}{2}(x(\delta_E(s)) - x(\text{twinkl}(s))) \geq \frac{1}{2}$, and this gives the desired contradiction.

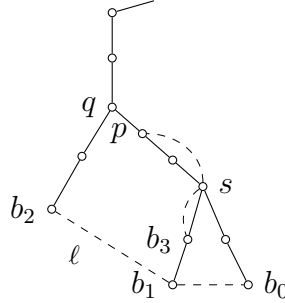


Figure 7.10: Illustration of a buddy link ℓ in the proof of Lemma 7.6.5. It shows that ℓ is overlapping with links in both $\delta_E^{\text{out}}(s)$ and $\delta_E(\hat{e}) \cap \delta_E(s)$, where \hat{e} is the tree-edge between b_3 and s . For example, ℓ is overlapping with the links sp and sb_3 .

For any M -covered leaf w in T'_v , by Lemma 7.5.1, w is an original node, and moreover, we have $x(\delta_E(w)) \leq 1$ by Lemma 6.4.3. Thus, we have $x(\delta_E(w)) \leq 1$ for each $w \in L^{\text{matched}}(T'_v) \cup (\mathcal{S} \cap V(T'_v)) = \text{ML}(T'_v)$.

Let $\tilde{J} = \bar{J} - \bigcup_{w \in \text{ML}(T'_v)} \delta_E(w)$; this is the set of links in \bar{J} but not in $\text{AG}(T'_v)$. We have $x(\tilde{J}) < 1$; otherwise, by Lemma 7.6.4, we would have $\Phi(x, T'_v) \geq \min\{\frac{1}{2}, \frac{1}{2}x(\tilde{J})\} \geq \frac{1}{2}$, which is a contradiction. We have

$$\begin{aligned} 1 > x(\tilde{J}) &= x(\bar{J}) - \sum_{w \in \text{ML}(T'_v)} x(\bar{J} \cap \delta_E(w)) \geq |\text{AU}(T'_v)| - |\text{ML}(T'_v)| \\ &= |U(T'_v)| + 1 - 2|M(T'_v)| - |\mathcal{S} \cap V(T'_v)| \\ &= |U(T'_v)| - (1 + |\mathcal{S} \cap V(T'_v)|) \geq 0. \end{aligned}$$

Hence, we have $|U(T'_v)| = |\mathcal{S} \cap V(T'_v)| + 1$, and $|\text{ML}(T'_v)| = |\text{AU}(T'_v)|$. For each $w \in \text{ML}(T'_v)$, we have $x(\bar{J} \cap \delta_E(w)) = 1$ because $x(\ell)$ is integral for each link $\ell \in \bar{J} \cap \delta_E(w)$ (otherwise, $\sum_{w \in \text{ML}(T'_v)} x(\bar{J} \cap \delta_E(w)) \leq |\text{ML}(T'_v)| - 1$ implies that $x(\tilde{J}) \geq |\text{AU}(T'_v)| - |\text{ML}(T'_v)| + 1 \geq 1$).

For each $w \in \text{ML}(T'_v)$, define ℓ_w to be the link in $\delta_E(w) \cap \bar{J}$ with $x(\ell_w) = 1$. We claim that these links form a perfect matching of the auxiliary graph, thus, $\text{AM}(T'_v) = \{\ell_w : w \in \text{ML}(T'_v)\}$. Otherwise, there exist two links ℓ_{w_1}, ℓ_{w_2} that are incident to the same node $u \in \text{AU}(T'_v)$. Then, $x(\bar{J}) \geq |\text{AU}(T'_v)| + 1$, which implies that $x(\tilde{J}) = x(\bar{J}) - \sum_{w \in \text{ML}(T'_v)} x(\bar{J} \cap \delta_E(w)) \geq |\text{AU}(T'_v)| + 1 - |\text{ML}(T'_v)| \geq 1$. This contradicts the fact that $x(\tilde{J}) < 1$. Our claim follows.

We claim that $\text{AM}(T'_v)$ covers T'_v . Otherwise, there exists a tree-edge \hat{e}_0 in T'_v that is not covered by $\text{AM}(T'_v)$. Let $\delta_E^+(\hat{e}_0)$ denote the set of links of positive x -value in $\delta_E(\hat{e}_0)$. Then, we have $x(\delta_E^+(\hat{e}_0)) \geq 1$, and none of the links in $\delta_E^+(\hat{e}_0)$ is incident to $(\mathcal{S} \cap V(T'_v)) \cup L^{\text{matched}}(T'_v)$; the latter assertion holds because $x(\ell_w) = 1$ and $x(\delta_E(w)) \leq 1$ for every $w \in \text{ML}(T'_v)$, i.e., the nodes in $\text{ML}(T'_v)$ are already “saturated” by $\text{AM}(T'_v)$. Thus, Lemma 7.6.4 applies to $\delta_E^+(\hat{e}_0)$, and we have $\Phi(x, T'_v) \geq \min\{\frac{1}{2}, \frac{1}{2}x(\delta_E^+(\hat{e}_0))\} \geq \frac{1}{2}$, which is a contradiction. Our claim follows: $\text{AM}(T'_v)$ covers T'_v .

Additionally, we claim that $\text{AM}(T'_v)$ has no links between \bar{v} and $\mathcal{S} \cap V(T'_v)$. By way of contradiction, assume that there exists a link $\ell = sq \in \text{AM}(T'_v)$ such that $s \in \mathcal{S} \cap V(T'_v)$ and $q \notin V(T'_v)$. Suppose that q is an original non-leaf node. Then sq is a link between two original non-leaf nodes with $x(sq) = 1$, hence, we have $\Phi(x, T'_v) \geq \frac{1}{2}$ due to Fact 7.3.2 and the following term in $\Phi(x, T'_v)$:

$$\frac{1}{2}x(E((V-L) \cap V(T'_v), (V-L) \cap (V(T') - V(T'_v)))).$$

This is a contradiction. Otherwise, if q is a compound node or an original leaf, then we claim that an (up-to-5) greedy contraction applies, and this too gives a contradiction. Note that $|\mathcal{S} \cap V(T'_v)| \leq 2$ and $|M(T'_v)| \leq 1$, hence, $|\text{AM}(T'_v)| \leq 4$. The credit assigned to the

leaves and matching links in T'_v is $\frac{3}{2} + |U(T'_v)| = \frac{3}{2} + |\mathcal{S} \cap V(T'_v)| + 1 = |AM(T'_v)| + \frac{1}{2}$. If q is a compound node or an original M -exposed leaf, then q provides one additional credit, and this suffices for an (up-to-5) greedy contraction of the links of $AM(T'_v)$; otherwise, q is an M -covered leaf, and the M -link ℓ_q incident to q provides additional $\frac{3}{2}$ credit, and this suffices for an (up-to-5) greedy contraction of ℓ_q together with the links of $AM(T'_v)$. Our claim follows: $AM(T'_v)$ has no link of the form $\bar{v}s$, $s \in \mathcal{S} \cap V(T'_v)$.

Finally, we prove that $\text{twinlk}(s) = uw$ is in $AM(T'_v)$ for each stem $s \in \mathcal{S} \cap V(T'_v)$. By Lemma 7.6.1, T'_s is completely contained in T'_v and every node in T'_s is original. We first claim that no buddy links associated with s have positive x -value if they exist. Otherwise, without loss of generality, suppose that a buddy link $l_u = uq$ in the original input link set E associated with s has positive x -value where both u, q are original ends. Since T'_v has at least 3 leaves and $C(T'_v) \neq \emptyset$, we have if q is contained in some compound node, then this compound node must be a leaf in T'_v . If q is not contained in a compound node, then q is an original leaf in T'_v as well since T'_v has at least 3 leaves. Consider the link l_s in $AM(T'_v)$ incident with s . If the end of l_s other than s is not in T_s , then it will be overlapping with l_u and $x(l_u) > 0, x(l_s) = 1$. This contradicts the overlapping constraints in LP7. By the similar argument, we know l_s cannot be us . Thus, l_s must be ws . Let p be the least common ancestor of u and q (see Figure 7.11).

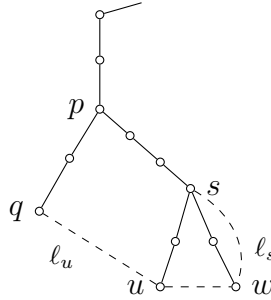


Figure 7.11: The links ℓ_u and ℓ_s in the proof of Lemma 7.6.5.

Consider the link $\bar{\ell}$ in $AM(T'_v)$ covering the tree-edge between p and its parent (p can be v ; in this case, $\bar{\ell}$ is the link incident with \bar{v} in $AM(T'_v)$). Note that $\bar{\ell}$ has an end at a leaf in T'_p . Then $\bar{\ell}$ will be either overlapping with l_s or l_u . This is a contradiction to the overlapping constraints in LP7. Thus, no buddy links associated with s have positive x -value if they exist. Note that one of u, w is an M -covered leaf due to Lemma 7.5.1. Thus, if the twin link uw is not in $AM(T'_v)$, then $x(uw) = 0$. Then, we have $\Phi(x, T'_v) \geq \text{slack}_x(s) \geq \frac{1}{2}x(l_s) = \frac{1}{2}$. This is a contradiction.

□

Theorem 7.6.6 *Suppose that no (up-to-5) greedy contractions are applicable. Let T'_v be a semi-closed tree that is not good. Then one of the following holds for T'_v .*

1. T'_v is a deficient 3-leaf tree.
2. T'_v is a deficient 4-leaf tree.
3. T'_v has 4 leaves with $|M(T'_v)| = 1$, and moreover T'_v has no cover of size 3.

Proof. Since T'_v is not good, Lemma 7.5.2 and Lemma 7.6.5 (1) imply that $T'_v \neq T'$, $C(T'_v) = \emptyset$, and $|M(T'_v)| = 1$.

Observe that T'_v has at least one M -exposed leaf. Otherwise, since $|M(T'_v)| = 1$, T'_v has exactly two leaves and there is an M -link between these two leaves; moreover, by Lemma 7.5.1(1), every node on the path of T' between these two leaves is original; it follows that the link in $M(T'_v)$ is a twin link; this contradicts the definition of M . Since T'_v has an M -exposed leaf and exactly two M -covered leaves, it follows that T'_v has at least three leaves.

Also, observe that $|\mathcal{S} \cap V(T'_v)| \leq 2$. Otherwise, suppose that $|\mathcal{S} \cap V(T'_v)| \geq 3$. Then, by Lemma 7.6.1, for every stem $s \in \mathcal{S} \cap V(T'_v)$, every node in T'_s is original, hence, there exists a stem $s^* \in \mathcal{S} \cap V(T'_v)$ such that both the leaves of T'_{s^*} are M -exposed and there exists a twin link between these two leaves; this contradicts Lemma 7.5.1(2).

Let \hat{e} denote the tree-edge between v and its parent. Let J denote $\{\ell \in \delta_E(u) : u \in (\mathcal{S} \cap V(T'_v)) \cup L^{\text{matched}}(T'_v) \cup \mathcal{B}(T'_v)\}$, where $\mathcal{B}(T'_v) = \mathcal{B}^{\text{orig}}(T'_v) \cup \mathcal{B}^{\text{comp}}(T'_v)$ (see the discussion before Lemma 7.6.3 for the definitions of $\mathcal{B}^{\text{orig}}(T'_v)$, $\mathcal{B}^{\text{comp}}(T'_v)$).

By Lemma 7.6.3, y can be written as a convex combination $\sum_{i \in Z} \lambda_i x^i$ such that $x^i \in \text{Las}^3_{\text{proj}}(\text{LP7})$ and $x^i|_J$ is integral. Since T'_v is not good and $|M(T'_v)| = 1$, Lemma 7.5.2 implies that $\Phi(y, T'_v) < \frac{1}{2}$. Thus, there exists an $i_0 \in Z$ such that $\Phi(x^{i_0}, T'_v) < \frac{1}{2}$.

We claim that $|U(T'_v)| \leq |\mathcal{S} \cap V(T'_v)| + 1$. To see this, suppose that $|U(T'_v)| \geq |\mathcal{S} \cap V(T'_v)| + 1$. Then, all of the conditions of Lemma 7.6.5(2) apply, hence, the lemma implies that $|U(T'_v)| = |\mathcal{S} \cap V(T'_v)| + 1$. Our claim follows.

Now, we analyze a few cases, depending on the number of leaves of T'_v . By Lemma 7.6.1, for every stem $s \in \mathcal{S} \cap V(T'_v)$, every node in T'_s is original; thus, each stem $s \in \mathcal{S} \cap V(T'_v)$ contributes two original leaves to T'_v . Therefore, $|\mathcal{S} \cap V(T'_v)| \leq |L(T'_v)|/2$.

Case 1 T'_v has exactly three leaves. Let the three leaves be a, b_1, b_2 , where a is M -exposed and b_1, b_2 are M -covered; thus, b_1b_2 is the unique link in $M(T'_v)$, and b_1, b_2 are original nodes by Lemma 7.5.1. Clearly, $|\mathcal{S} \cap V(T'_v)| \leq 1$. We have two subcases, depending on $|\mathcal{S} \cap V(T'_v)|$.

Subcase 1.1 $|\mathcal{S} \cap V(T'_v)| = 1$. Let s be the stem in T'_v . The M -link b_1b_2 cannot have both ends in T'_s (since M contains no twin links). Thus, the M -exposed leaf a is a leaf of T'_s . Then, it can be seen that a is a bud with a buddy link b_1b_2 . This is a contradiction, since M contains no buddy links.

Subcase 1.2 $|\mathcal{S} \cap V(T'_v)| = 0$. Then we have $|U(T'_v)| = 1 = 1 + |\mathcal{S} \cap V(T'_v)|$. By Lemma 7.6.5, there exist two links $\ell_v \in \delta_E(\hat{e})$ and $\ell_a \in \delta_E(a)$ such that $x^{i_0}(\ell_v) = x^{i_0}(\ell_a) = 1$, these two links cover T'_v , and moreover, each of b_1, b_2 is incident to exactly one of these two links (since the auxiliary graph has a perfect matching formed by these two links).

If there is only one non-leaf node with degree other than 2 in T'_v (see Figure 7.6(a)), then T'_v is a deficient 3-leaf tree. We are done. Otherwise, we have exactly two non-leaf nodes u, q in T'_v with degree other than 2. In fact, both these two nodes have exactly degree 3 since T'_v has exactly 3 leaves. Without loss of generality, we assume u is an ancestor of q . Then, T'_q has only two leaves. By the argument at the beginning of the proof, the M -link in T'_v cannot connect both leaves in T'_q . This implies that one leaf of T'_q is M -exposed. So, it is a . Without loss of generality, we can assume that the other leaf of T'_q is b_2 . Then, b_1 is the third leaf, which is not in T'_q .

Suppose that ℓ_v is incident to b_1 and ℓ_a is incident to b_2 . Then, the tree-edge between q and its parent is not covered by these two links (see Figure 7.12(a)). This is a contradiction. Hence, ℓ_v is incident to b_2 and ℓ_a is incident to b_1 (see Figure 7.12(b)). Therefore, T'_v satisfies all the conditions of a deficient 3-leaf tree.

Case 2 T'_v has exactly four leaves. Let the four leaves be a_1, a_2, b_1, b_2 , where a_1, a_2 are M -exposed and b_1, b_2 are M -covered; thus, b_1b_2 is the unique link in $M(T'_v)$, and b_1, b_2 are original nodes by Lemma 7.5.1. We have $1 = |U(T'_v)| - 1 \leq |\mathcal{S} \cap V(T'_v)| \leq 2$. As above, we have two subcases, depending on $|\mathcal{S} \cap V(T'_v)|$.

Subcase 2.1 $|\mathcal{S} \cap V(T'_v)| = 1$. Let s be the stem in T'_v . By Lemma 7.5.1, it can be seen that the twin link of s is incident to one M -exposed leaf, say a_1 , and to one M -covered leaf, say b_1 (see Subcase 1.1).

Note that $|U(T'_v)| = 2 = 1 + |\mathcal{S} \cap V(T'_v)|$. By Lemma 7.6.5, there exist three links a_1b_1 (the twin link of s), $\ell_v \in \delta_E(\hat{e})$ and $\ell_{a_2} \in \delta_E(a_2)$ such that $x^{i_0}(a_1b_1) =$

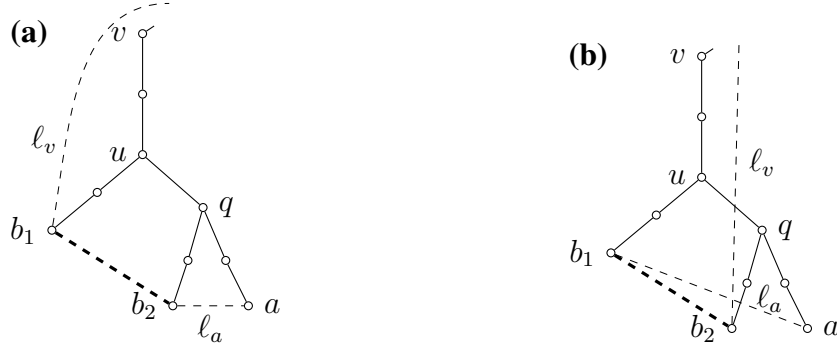


Figure 7.12: The links ℓ_v and ℓ_a in Subcase 1.2 of the proof of Theorem 7.6.6. The dashed lines indicate links and the thick dashed lines indicate M -links.

$x^{i_0}(\ell_v) = x^{i_0}(\ell_{a_2}) = 1$, these three links cover T'_v , and moreover, each of s, b_2 is incident to exactly one of the two links ℓ_v, ℓ_{a_2} (since the auxiliary graph has a perfect matching formed by the three links); moreover, s cannot be incident to ℓ_v (by Lemma 7.6.5(2)(iv)). Thus ℓ_v is incident to b_2 , and ℓ_{a_2} is incident to s , i.e., $\ell_{a_2} = a_2s$. Let p be the least common ancestor of s and a_2 (in T'_v). If T'_p does not contain all the leaves of T'_v , then it can be seen that the tree-edge between p and its parent is not covered by the three links a_1b_1, a_2s, ℓ_v (see Figure 7.13(a)). It follows that T'_p contains all the leaves of T'_v . Then, it can be seen that T'_v is a deficient 4-leaf tree (see Figure 7.13(b)).

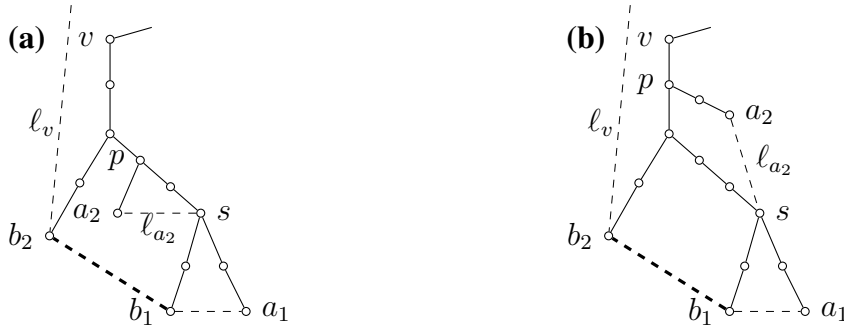


Figure 7.13: The links ℓ_v, ℓ_{a_2} and a_1b_1 in Subcase 2.1 of the proof of Theorem 7.6.6. The dashed lines indicate links and the thick dashed lines indicate M -links.

Subcase 2.2 $|\mathcal{S} \cap V(T'_v)| = 2$. If T'_v has no cover of size 3, then item (3) in the statement of the theorem holds. Thus, we may assume that T'_v has a cover \bar{J} with $|\bar{J}| = 3$.

Let s_1, s_2 be the two stems in T'_v . Let a_1, b_1 be the original leaves of T'_{s_1} , and let a_2, b_2 be the original leaves of T'_{s_2} . We may assume that a_1, a_2 are M -exposed and b_1, b_2 are M -covered, because each twin link is incident to one M -exposed leaf and to one M -covered leaf by Lemma 7.5.1 (see Subcase 2.1). Thus, b_1b_2 is the unique link in $M(T'_v)$.

By Lemma 7.6.1 and $C(T'_v) = \emptyset$, T'_v has no compound nodes, i.e., all its nodes are original.

Observe that T'_v has at least $3\frac{1}{2}$ credits from the two M -exposed leaves and the M -link.

Consider the possible leaf-to-leaf links of T'_v . The link a_1a_2 does not exist, by Lemma 7.5.1, since a_1, a_2 are both M -exposed. If both the links a_1b_2 and a_2b_1 exist, then an (up-to-5) greedy contraction applies (we contract these two links and we have $3\frac{1}{2}$ credits). Thus, at most one of the links a_1b_2 or a_2b_1 is present.

If T_v has no leafy 3-cover in T , then it satisfies all the conditions for a bad 2-stem tree, hence, it would have been contracted in Preprocessing step 1 (Λ -contraction). This is a contradiction.

Thus, we may assume that T_v in T has a leafy 3-cover $\tilde{J} = \{\ell_0, \ell_1, \ell_2\}$ where ℓ_0 is a link with ends u, w such that u is in T_v and w is a leaf in $L - L(T_v)$. Let $\ell'_0 = uw'$ be the corresponding link w.r.t. T' for ℓ_0 where $w' = w$ if w is still an original node in T' , and otherwise, w' is the compound node containing w . Note that in either of the cases, w' is not in $V(T'_v)$ since every node in T'_v is original. If w' is a compound node or an M -exposed original leaf, then an (up-to-5) greedy contraction applies (we contract the 3 links in \tilde{J} and we have $4\frac{1}{2}$ credits). Otherwise, if w' is an M -covered original leaf, then again an (up-to-5) greedy contraction applies (we contract 4 links, the links in \tilde{J} and the M -link incident to w , and we have $3\frac{1}{2} + \frac{3}{2} = 5$ credits). Thus, we get a contradiction from the existence of a leafy 3-cover.

Case 3 T'_v has at least five leaves. Then, T'_v has at least three M -exposed leaves and two M -covered leaves. Thus, we have $3 \leq |U(T'_v)| \leq |\mathcal{S} \cap V(T'_v)| + 1 \leq 3$, since $|\mathcal{S} \cap V(T'_v)| \leq 2$; it follows that $|U(T'_v)| = 3$, $|\mathcal{S} \cap V(T'_v)| = 2$, and T'_v has exactly 5 leaves.

Let s_1, s_2 be the two stems in T'_v . Let a_1, b_1 be the original leaves of T'_{s_1} , and let a_2, b_2 be the original leaves of T'_{s_2} . We may assume that a_1, a_2 are M -exposed and b_1, b_2 are M -covered, because each twin link is incident to one M -exposed leaf and to one M -covered leaf by Lemma 7.5.1 (see Subcase 2.1). Thus, b_1b_2 is the unique link in $M(T'_v)$. Let u be the fifth leaf of T'_v , where u is not in T'_{s_1} nor in T'_{s_2} .

Note that $|U(T'_v)| = 3 = 1 + |\mathcal{S} \cap V(T'_v)|$. By Lemma 7.6.5, there exist four links a_1b_1, a_2b_2 (the twin links of s_1, s_2), $\ell_v \in \delta_E(\hat{e})$ and $\ell_u \in \delta_E(u)$ such that each of s_1, s_2 is incident to

exactly one of the two links ℓ_v, ℓ_u (since the auxiliary graph has a perfect matching formed by the four links); moreover, ℓ_v cannot be incident to $\{s_1, s_2\}$ (by Lemma 7.6.5(2)(iv)). This is impossible. Thus, T'_v cannot have more than four leaves.

The result follows from the above case analysis. \square

7.6.3 Addressing deficient trees

For a deficient 3-leaf (4-leaf, respectively) tree T'_v , if T'_v is not a proper subtree of another deficient 3-leaf (4-leaf, respectively) tree, then we call T'_v a *maximal deficient 3-leaf (4-leaf, respectively) tree*. By Property 6.2.1, any two different maximal deficient 3-leaf (4-leaf, respectively) trees are disjoint.

To handle the deficient 3-leaf trees and deficient 4-leaf trees, we contract some special links and then replace some links in the matching M to form a new matching M^{new} . Specifically, we first contract the link set E^{latch} of latches in maximal deficient 4-leaf trees to form $T'' := T' \setminus E^{latch}$, and then we examine each maximal deficient 3-leaf tree T''_w in T'' and replace the unique link of $M(T''_w)$ by another leaf-to-leaf link. In more detail, consider any maximal deficient 3-leaf tree T''_w in T'' , and let the three leaves be a, b, d , where a is M -exposed, b is the ceiling leaf, and bd is the unique link in $M(T''_w)$; we keep the link ad in M^{new} instead of the M -link bd (see Algorithm 7.2). Since any two different maximal deficient 3-leaf trees are disjoint, this replacement takes place independently for each maximal deficient 3-leaf tree.

The new compound node in T'' formed by contracting a latch in T' is called *latched compound node*. Since two different maximal deficient 4-leaf trees are disjoint, this contraction of each latch takes place independently and every latch results in one latched compound node. For a subtree T''_v of T'' , we define $latch(T''_v) := \{\ell(\langle c^{latch} \rangle) : \langle c^{latch} \rangle \text{ is a latched compound node} \in T''_v\}$, where $\ell(\langle c^{latch} \rangle)$ is the latch associated with the latched compound node $\langle c^{latch} \rangle$.

We remark that the latched compound nodes are formed only for finding a good semiclosed tree of T' in the main loop of the algorithm. They are not the real compound nodes formed by contracting links in F . In other words, the latches are not added to F just due to these latched compound nodes to form T'' . Hence, there is no need to consider credit assignment for the contraction of latches.

If we contract the latch cs in a deficient 4-leaf tree T'_v where c is a leaf and s is a stem, then the resulting tree $T''_{v'}$ is a deficient 3-leaf tree (see Figure 7.7) where v' is the corresponding node in T'' to v in T' . When v is not on the path in T' for the link cs , then $v' = v$; otherwise, v' is the latched compound node formed by contracting cs . The links $ab_1, up(b_2)b_2$ witness the deficiency

of T''_v where a, b_1 are leaves of T'_s and b_1b_2 is the M -link in T'_v . Let b be the ceiling leaf of T''_v . Then, $up(b)$ is an ancestor of $up(b_2)$ in T'' by the definition of ceiling leaf.

Theorem 7.6.7 *Suppose that no (up-to-5) greedy contractions are applicable. Let T''_v be a minimally semiclosed tree in $T'' = T' \setminus E^{latch}$ w.r.t. M^{new} . Then, v cannot be a latched compound node in T'' (implying that T'_v is well defined). Furthermore, T'_v is a good semiclosed tree w.r.t. M and T'_v has a fitting cover $\Gamma(M^{new}, T''_v) \cup latch(T''_v)$ of size $|\Gamma(M, T'_v)|$.*

Proof. Suppose v is a latched compound node. Then, T''_v is a subtree of a deficient 3-leaf tree in T'' that formed by contracting the corresponding latch in a deficient 4-leaf tree in T' . Hence, T''_v is a subtree of a maximal deficient 3-leaf tree in T'' . However, by how we form M^{new} , any subtree of a maximal deficient 3-leaf tree is not semiclosed w.r.t. M^{new} . This is because the ceiling leaf of the maximal deficient 3-leaf tree is M^{new} -exposed. Thus, T''_v is not semiclosed w.r.t. M^{new} . This is a contradiction. Hence, v cannot be a latched compound node. This implies that v is a node in T' and T'_v is well defined.

Claim 7.6.8 *If T'_v has a node in a maximal deficient 4-leaf tree in T' , then T'_v properly contains this maximal deficient 4-leaf tree. If T'_v has a node in a maximal deficient 3-leaf tree in T' , then T'_v properly contains this maximal deficient 3-leaf tree.*

Suppose T'_v has a node in a maximal deficient 4-leaf tree T'_w . Since contracting the latch in T'_w results into a deficient 3-leaf tree T''_w in T'' , Thus, T''_v has a node in a maximal deficient 3-leaf tree T''_u where u is an ancestor of w' in T'' . However, every subtree of T''_u is not semiclosed w.r.t. M^{new} , which implies that T''_v is not a subtree of T''_u . By Property 6.2.1, we have T''_v properly contains T''_u . Hence, v is a proper ancestor of w in T' and T'_v properly contains T'_w .

Suppose T'_v has a node in a maximal deficient 3-leaf tree T'_w . If this maximal deficient 3-leaf tree is contained in some maximal deficient 4-leaf tree, then we are done by the analysis above. Hence, we can assume T'_w has no node in a maximal deficient 4-leaf tree by Property 6.2.1. Then $T''_w = T'_w$ is still a maximal deficient 3-leaf tree in T'' and no node in T''_w is a latched compound node. Then, T''_v has a node in T''_w . Similarly, T'_w is properly contained in T'_v . This proves the claim.

Claim 7.6.8 implies T'_v is neither a deficient 3-leaf tree nor a deficient 4-leaf tree in T' .

Let u be an M -exposed leaf of T'_v . If u belongs to some deficient 3-leaf tree or deficient 4-leaf tree T'_w , then T'_v contains this T'_w by Claim 7.6.8. Since T'_w is a semiclosed tree w.r.t. M in T' , then all links that incident with u have both ends in T'_w . Thus, all links that incident with u have both ends in T'_v . If u does not belong to any deficient 3-leaf tree or deficient 4-leaf tree,

then u is M^{new} -exposed as well in T'' . Since T_v'' is semiclosed w.r.t. M^{new} , we also have all links that incident with u have both ends in T_v' . Note that the replacement of M -links only takes place locally in a maximal deficient 3-leaf tree or a maximal deficient 4-leaf tree in T' . Thus, T_v' is semiclosed w.r.t. M , and the number of M -links in T_v' is same as the number of M^{new} -links in it. Thus, $|\Gamma(M, T_v')| = |\Gamma(M^{new}, T_v'')| + |latch(T_v'')|$.

Since T_v' is a minimally semiclosed tree w.r.t. M^{new} , by Lemma 6.6.2, $\Gamma(M^{new}, T_v'')$ is a fitting cover of T_v'' . This implies that $\Gamma(M^{new}, T_v'') \cup latch(T_v'')$ is a fitting cover of T_v' of size $|\Gamma(M, T_v')|$. Suppose T_v' is not good. Since T_v' is neither a deficient 3-leaf tree nor a deficient 4-leaf tree in T' , by Theorem 7.6.6, we have T_v' has 4 leaves, $|M(T_v')| = 1$, and T_v' has no cover of size 3. However, in this case, $|\Gamma(M, T_v')| = 3$. This is a contradiction. Therefore, T_v' is good. This completes the proof. \square

The procedure described above to find a good semiclosed tree is summarized as the following pseudocode.

Algorithm 7.2: Find a good semiclosed tree by addressing deficient trees.

```

1 for each maximal deficient 4-leaf tree  $T_w'$  in  $T'$  do
2   | contract the latch of  $T_w'$ ;
3 end
4 let  $T''$  be the resulting tree;
5 start with  $M^{new} := M$ ;
6 for each maximal deficient 3-leaf tree  $T_w''$  in  $T''$  do
7   | let  $b$  be the ceiling leaf,  $a$  be the  $M$ -exposed leaf and  $db$  be the  $M$ -link in  $T_w''$ ;
8   | update  $M^{new}$  by replacing  $db$  by  $da$  ( $M^{new} := M^{new} - \{db\} \cup \{da\}$ );
9 end
10 find a minimally semiclosed tree  $T_v''$  w.r.t.  $M^{new}$  (note that  $M^{new}$  is a matching of the
    leaf-to-leaf links);
11  $T_v'$  is a good semiclosed tree w.r.t.  $M$  with a fitting cover  $\Gamma(M^{new}, T_v'') \cup latch(T_v'')$  of size
     $|\Gamma(M, T_v')|$  by Theorem 7.6.7;
```

Algorithm 7.2 shows how to find a good semiclosed tree T_v' with a fitting cover of size $|\Gamma(M, T_v')|$ for the main loop in Algorithm 7.1 in Section 7.5. The contractions of latches to form T'' are only for this purpose. They are not counted for F in Algorithm 7.1.

In conclusion, Algorithm 7.1 runs in polynomial time and returns a solution for TAP with size at most $(\frac{3}{2} + \epsilon)y(E)$. This proves Theorem 7.4.1.

Chapter 8

Conclusion

In this chapter, we summarize the main results in this thesis and pose some open questions for future work.

8.1 Path TSP

In Chapter 3, we design a simple LP-based 1.5-approximation algorithm for the s - t path graph-TSP. For the more general metric s - t path TSP, in Chapter 4, we present a simple, self-contained analysis that unifies the results of [1] and [61]; our main contribution is a unified correction vector. A unified fractional T -join (see Section 4.2) is constructed based on this unified correction vector. Then, the main results of [1] and [61] can be presented as two different analyses of the cost of the same unified fractional T -join for the randomized Christofides' algorithm. An immediate question is whether we could have a better analysis of the cost of this unified fractional T -join. If true, this may imply an improved approximation guarantee for the randomized Christofides' algorithm.

For the metric s - t path TSP, a big open question is whether there exists a 1.5-approximation algorithm. If the 1.5-approximation factor is achieved, then this would close the gap and match the known lower bound on the integrality ratio of the path Held-Karp LP relaxation. The randomized Christofides' algorithm is the first one to surpass the long standing $\frac{5}{3}$ -approximation factor. The instance in Section 4.4 shows the crucial role of randomness for improving the approximation factor. Specifically, the randomized Christofides' algorithm samples a spanning tree based on a probability distribution, which is generated by an arbitrary convex decomposition in the spanning tree polytope of an optimal solution of the path Held-Karp LP relaxation. One idea

is to elaborately choose a particular convex decomposition to form a probability distribution with the aim of improving the cost analysis of the algorithm. In fact, very recently, Vygen [67] employed this idea by reassembling spanning trees and slightly improved the approximation factor to 1.5999.

8.2 Asymmetric TSP

In Chapter 5, we prove that the integrality ratio for level t of the Sherali-Adams system starting with the standard LP (DFJ LP) relaxation of ATSP is $\geq 1 + \frac{1-\epsilon}{2t+3}$ for any fixed integer $t \geq 0$ and small ϵ , $0 < \epsilon \ll 1$. To obtain this lower bound on the integrality ratio, we construct a fractional feasible solution for level t of the SA system. Unfortunately, the solution given by our construction is not positive semidefinite; thus, it does not apply to the LS_+ system or to the Las system. One natural question is whether there exists a fractional feasible solution of the LS_+ system that can be used to show a lower bound larger than 1 on the integrality ratio.

To construct a feasible solution of the stronger Las system is more difficult. This is due to the fact that the Las system not only requires positive semidefiniteness but also places global constraints via \mathcal{P}_{2t+2} (e.g., $M_{t+1}(y) \succeq 0$). Mastrolilli [52] showed a large integrality ratio for the Las system applied to Capacitated Covering Problems. One key technique there is to force the constant term b of some constraint $a^T x \geq b$ to be sufficiently small. However, for the natural LP relaxations of TSP and its variants, the constant terms of constraints are fixed. There are no lower bound results known for ATSP for any of the Lift-and-Project systems based on positive semidefiniteness. We mention that for TSP, Cheung [18] showed an integrality ratio of $\frac{4}{3}$ for $O(1)$ levels of the LS_+ system. At level 0, it is well known that any integrality ratio for the standard LP relaxation for TSP applies also to ATSP. Unfortunately, this does not hold for level 1 or higher. Consequently, Cheung's results [18] for TSP do not apply to ATSP.

Our lower bound $1 + \frac{1-\epsilon}{2t+3}$ on the integrality ratio fade out as the level of the SA tightening increases, and for $t \geq 35$ (roughly) our integrality ratio falls below the hardness threshold of $\frac{75}{74}$ of [42]. Thus, our integrality ratio cannot be optimal, and it is possible that a large constant lower bound (e.g. 2) on the integrality ratio survives for $O(1)$ levels of the SA system. One research direction is to study whether such a constant exists. This is true for $O(1)$ levels of the LS_+ system starting with the standard LP relaxation of TSP [18].

8.3 TAP

The main result of Chapter 7 is to prove an upper bound of $(\frac{3}{2} + \epsilon)$ on the integrality ratio of a SDP relaxation, where $\epsilon > 0$ can be any small constant, by analyzing a combinatorial algorithm. This SDP relaxation is derived by applying level t of the **LAS** system to the initial LP relaxation **LP7** where $t = \max\{17, \lceil \frac{1}{2\epsilon} \rceil + 1\}$. Furthermore, there is a polynomial-time algorithm that always outputs a feasible solution of TAP of size $\leq (\frac{3}{2} + \epsilon)y(E)$, where y is an optimal solution of $\text{Las}_{proj}^t(\text{LP7})$.

We propose two open questions on TAP. One is to design a polynomial-time algorithm with an approximation factor better than $\frac{3}{2}$. The tight example in Section 6.8 shows that $\frac{3}{2}$ is the best approximation factor that we can achieve if we only employ contractions of semiclosed tree and greedy contractions of small link sets shown in Chapters 6 and 7. However, a different type of algorithm may exist to break the barrier. The other open question is on the weighted Tree Augmentation Problem. It is not hard to see that our LP relaxation with overlapping constraints can be extended to the weighted case. The best known approximation factor for the weighted Tree Augmentation Problem is 2, and a key open question is to improve on this approximation factor. One research direction is to study the integrality ratio of the **LAS** system applied to the weighted version of our LP relaxation of TAP. An open question is whether the integrality ratio is less than 2 for some constant level of the **LAS** system.

References

- [1] H-C. An, R. Kleinberg, and D. B. Shmoys. Improving Christofides' algorithm for the s-t path TSP. In *Proceedings of the 44th Annual ACM symposium on Theory of Computing (STOC)*, pages 875–886, 2012.
- [2] H-C. An and D. B. Shmoys. LP-based approximation algorithms for Traveling Salesman Path Problems. *CoRR*, abs/1105.2391, 2011.
- [3] N. Anari and S. O. Gharan. Effective-resistance-reducing flows and Asymmetric TSP. *CoRR*, abs/1411.4613, 2014.
- [4] S. Arora, B. Bollobás, L. Lovász, and I. Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(1):19–51, 2006.
- [5] A. Asadpour, M. X. Goemans, A. Madry, S. O. Gharan, and A. Saberi. An $O(\log n / \log \log n)$ -approximation algorithm for the Asymmetric Traveling Salesman Problem. In *Proceedings of the 21th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 379–389, 2010.
- [6] Y-H. Au and L. Tunçel. Complexity analyses of Bienstock-Zuckerberg and Lasserre relaxations on the matching and stable set polytopes. In *Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 14–26, 2011.
- [7] S. Benabbas, S. O. Chan, K. Georgiou, and A. Magen. Tight gaps for vertex cover in the Sherali-Adams SDP hierarchy. In *Proceedings of the 31st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 41–54, 2011.
- [8] D. Bertsimas and C-P. Teo. The parsimonious property of cut covering problems and its applications. *Operations Research Letters*, 21(3):123–132, 1997.

- [9] R. Carr and S. Vempala. On the Held-Karp relaxation for the asymmetric and symmetric Traveling Salesman Problems. *Mathematical Programming*, 100(3):569–587, July 2004.
- [10] M. Charikar, M. X. Goemans, and H. J. Karloff. On the integrality ratio for the Asymmetric Traveling Salesman Problem. *Mathematics of Operations Research*, 31(2):245–252, 2006.
- [11] M. Charikar, K. Makarychev, and Y. Makarychev. Integrality gaps for Sherali-Adams relaxations. In *Proceedings of the 41st Annual ACM symposium on Theory of Computing (STOC)*, pages 283–292, 2009.
- [12] J. Cheriyan and Z. Gao. Approximating (Unweighted) Tree Augmentation via Lift-and-Project, Part I. *To appear on arXiv*.
- [13] J. Cheriyan and Z. Gao. Approximating (Unweighted) Tree Augmentation via Lift-and-Project, Part II. *CoRR*, abs/1507.01309, 2015.
- [14] J. Cheriyan, Z. Gao, K. Georgiou, and S. Singla. On integrality ratios for Asymmetric TSP in the Sherali-Adams hierarchy. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 340–351, 2013.
- [15] J. Cheriyan, Z. Gao, K. Georgiou, and S. Singla. On integrality ratios for Asymmetric TSP in the Sherali-Adams hierarchy. *CoRR*, abs/1405.0945, 2014.
- [16] J. Cheriyan, T. Jordán, and R. Ravi. On 2-coverings and 2-packings of laminar families. In *Algorithms - ESA '99, 7th Annual European Symposium*, volume 1643 of *Lecture Notes in Computer Science*, pages 510–520. Springer, 1999. A longer version is on the web: <http://www.math.uwaterloo.ca/~jcheriyan/publications.html>.
- [17] J. Cheriyan, H. J. Karloff, R. Khandekar, and J. Könemann. On the integrality ratio for tree augmentation. *Operations Research Letters*, 36(4):399–401, 2008.
- [18] K. K. H. Cheung. On Lovász-Schrijver lift-and-project procedures on the Dantzig-Fulkerson-Johnson relaxation of the TSP. *SIAM Journal on Optimization*, 16(2):380–399, 2005.
- [19] E. Chlamtac and M. Tulsiani. Convex relaxations and integrality gaps. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on Semidefinite, Conic and Polynomial Optimization*, volume 166 of *International Series in Operations Research & Management Science*, pages 139–169. Springer US, 2012.
- [20] N. Christofides. Worst-case analysis of a new heuristic for the Travelling Salesman Problem. Technical Report 388, Graduate School of Industrial Administration, CMU, 1976.

- [21] N. Cohen and Z. Nutov. A $(1 + \ln 2)$ -approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius. *Theoretical Computer Science*, 489-490:67–74, 2013.
- [22] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley-Interscience, 1998.
- [23] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale Traveling Salesman Problem. *Journal of the Operations Research Society of America*, 2:393–410, 1954.
- [24] W. Fernandez de-la Vega and C. Kenyon-Mathieu. Linear programming relaxations of maxcut. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 53–61, 2007.
- [25] R. Diestel. *Graph Theory (4th ed.)*. Graduate Texts in Mathematics, Volume 173. Springer-Verlag, Heidelberg, 2010.
- [26] J. Edmonds and E. L. Johnson. Matching: A well-solved class of integer linear programs. In *Combinatorial Optimization*, pages 27–30, 2001.
- [27] P. Elliott-Magwood. *The Integrality Gap of the Asymmetric Travelling Salesman Problem*. PhD thesis, Department of Mathematics and Statistics, University of Ottawa, 2008.
- [28] G. Even, J. Feldman, G. Kortsarz, and Z. Nutov. A $3/2$ -approximation algorithm for augmenting the edge-connectivity of a graph from 1 to 2 using a subset of a given edge set. In *APPROX-RANDOM: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 4th International Workshop, APPROX 2001, and 5th International Workshop, RANDOM 2001, Proceedings*, pages 90–101, 2001.
- [29] G. Even, J. Feldman, G. Kortsarz, and Z. Nutov. A 1.8 approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2. *ACM Transactions on Algorithms*, 5:21:6–21:17, 2009.
- [30] S. Fiorini and N. Mutsanas. Notes on the tree augmentation problem. *Personal Communication*, 2011.
- [31] A. Frank. On a theorem of Mader. *Discrete Mathematics*, 101(1-3):49–57, 1992.
- [32] G. N. Frederickson and J. Ja’Ja’. Approximation algorithms for several graph augmentation problems. *SIAM Journal on Computing*, 10(2):270–283, 1981.

- [33] Z. Gao. An LP-based $3/2$ -approximation algorithm for the s - t path graph Traveling Salesman Problem. *Operations Research Letters*, 41(6):615–617, 2013.
- [34] Z. Gao. On the metric s - t path Traveling Salesman Problem. *CoRR*, abs/1404.7569, 2014.
- [35] K. Georgiou. Tree augmentation. *Personal Communication*, 2011.
- [36] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
- [37] M. Held and R. M. Karp. The traveling-salesman and minimum cost spanning trees. *Operations Research*, 18:1138–1162, 1970.
- [38] M. Held and R. M. Karp. The traveling-salesman problem and minimum spanning trees: part II. *Mathematical Programming*, 1:6–25, 1971.
- [39] J. A. Hoogeveen. Analysis of Christofides’ heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10:291–295, 1991.
- [40] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.
- [41] A. Karlin, C. Mathieu, and C. Nguyen. Integrality gaps of linear and semi-definite programming relaxations for knapsack. In *Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 301–314. 2011.
- [42] M. Karpinski, M. Lampis, and R. Schmied. New inapproximability bounds for TSP. In *Proceedings of the 24th International Symposium on Algorithms and Computation (ISAAC)*, volume 8283 of *Lecture Notes in Computer Science*, pages 568–578. Springer, 2013.
- [43] G. Kortsarz, R. Krauthgamer, and J. R. Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM Journal on Computing*, 33(3):704–720, 2004.
- [44] G. Kortsarz and Z. Nutov. A simplified 1.5 -approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2. *CoRR*, abs/1507.02799, 2015.
- [45] M. Lampis. Improved inapproximability for TSP. In *APPROX-RANDOM: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012. Proceedings*, pages 243–253, 2012.

- [46] J. B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM Journal on Optimization*, 12(3):756–769, 2002.
- [47] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, 28(3):470–496, 2003.
- [48] L. Lovász. Lecture. Conference of Graph Theory. Prague, 1974.
- [49] L. Lovász. Combinatorial problems and exercises. North-Holland, 1979.
- [50] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.
- [51] Y. Maduel and Z. Nutov. Covering a laminar family by leaf to leaf links. *Discrete Applied Mathematics*, 158(13):1424–1432, 2010.
- [52] M. Mastrolilli. The Lasserre hierarchy in almost diagonal form. *CoRR*, abs/1312.6493, 2013.
- [53] T. Mömke and O. Svensson. Approximating graphic TSP by matchings. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 560–569, 2011.
- [54] M. Mucha. 13/9-approximation for graphic TSP. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 30–41, 2012.
- [55] H. Nagamochi. An approximation for finding a smallest 2-edge connected subgraph containing a specified spanning tree. *Discrete Applied Mathematics*, 126:83–113, 2003.
- [56] C. H. Papadimitriou and S. Vempala. On the approximability of the Traveling Salesman Problem. *Combinatorica*, 26(1):101–120, 2006.
- [57] R. Roberti and P. Toth. Models and algorithms for the Asymmetric Traveling Salesman Problem: an experimental comparison. *EURO Journal on Transportation and Logistics*, 1:113–133, 2012.
- [58] T. Rothvoß. Directed Steiner tree and the Lasserre hierarchy. *CoRR*, abs/1111.5473, 2011.
- [59] T. Rothvoß. The Lasserre hierarchy in approximation algorithms. *MAPSP Tutorial: Lecture notes*, 2013. <http://www.math.washington.edu/~rothvoss/lecturenotes/lasserresurvey.pdf>.

- [60] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency, Algorithms and Combinatorics*, volume 24. Springer, Berlin, 2003.
- [61] A. Sebő. Eight-fifth approximation for the path TSP. In *Proceedings of the 16th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 362–374, 2013.
- [62] A. Sebő and J. Vygen. Shorter tours by nicer ears: $7/5$ -approximation for the graph-TSP, $3/2$ for the path version, and $4/3$ for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014.
- [63] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.
- [64] T. Stephen and L. Tunçel. On a representation of the matching polytope via semidefinite liftings. *Mathematics of Operations Research*, 24(1):1–7, 1999.
- [65] I. Turlakis. *New Lower Bounds for Approximation Algorithms in the Lovasz-Schrijver Hierarchy*. PhD thesis, Department of Computer Science, Princeton University, 2006.
- [66] J. Vygen. New approximation algorithms for the TSP. *Optima: Mathematical Optimization Society Newsletter*, 90(1-12), 2012.
- [67] J. Vygen. Reassembling trees for the traveling salesman. *CoRR*, abs/1502.03715, 2015.
- [68] T. Watson. Lift-and-project integrality gaps for the Traveling Salesperson Problem. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:97, 2011.